

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

HELPDESK PRO PODPORU SPRÁVY FIREMNÍCH POČÍTAČOVÝCH SÍTÍ

DIPLOMOVÁ PRÁCE

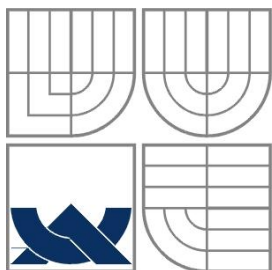
MASTER'S THESIS

AUTOR PRÁCE

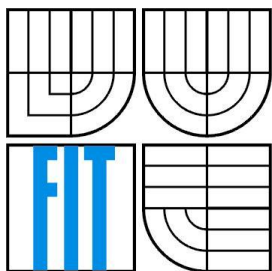
AUTHOR

Bc. PETR VAŠÁK

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

HELPDESK PRO PODPORU SPRÁVY FIREMNÍCH POČÍTAČOVÝCH SÍTÍ

HELPDESK FOR COMPUTER NETWORK MAINTENANCE SUPPORT

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. PETR VAŠÁK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAROSLAV RÁB

BRNO 2011

Abstrakt

Práce se zabývá vytvořením a implementací systému helpdesk s diagnostikou (zpětnou odezvou) pro firmu The Best Network Solution, zabývající se IT outsourcingem, sloužící ke správě jejich počítačových sítí, zařízení a podporu koncových zákazníků.

Abstract

The thesis is dealing with creating and implementation of helpdesk system with diagnostics (feedback) for The Best Network Solution, the IT outsourcing company, serving for maintenance theirs computer networks, equipment and end customers support.

Klíčová slova

Helpdesk, informační systém, správa sítí, správa počítačů, správa serverů, SharePoint, Asp.net, Visual Studio, C#, PowerShell, Windows Server

Keywords

Helpdesk, information system, network maintenance, computer maintenance, server maintenance, SharePoint, Asp.net, Visual Studio, C#, PowerShell, Windows Server

Citace

Bc. Petr Vašák: Helpdesk pro podporu správy firemních počítačových sítí, diplomová práce, Brno, FIT VUT v Brně, 2011

Helpdesk pro podporu správy firemních počítačových sítí

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Jaroslava Rába. Další informace mi poskytl ředitel firmy The Best Network Solution Ing. Igor Šimkovský. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Bc. Petr Vašák
23. 05. 2011

Poděkování

Děkuji Ing. Jaroslavu Rábovi za vedení celé práce a Ing. Igoru Šimkovskému za poskytnutí potřebných pracovních a výpočetních zdrojů.

© Bc. Petr Vašák, 2011

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah	1
1 Úvod	4
1.1 Popis jednotlivých kapitol	5
2 Firma, klienti a analýza požadavků	6
2.1 Seznámení s firmou BNS	6
2.1.1 Struktura firmy	6
2.2 Průzkum portfolia spravovaných firem	6
2.2.1 Druhy požadavků klientů	7
2.2.2 Delikomat	7
2.2.3 Super Zoo	9
2.2.4 AxFlow	10
2.3 Analýza síťových technologií a požadavků	11
2.3.1 Popis využívaných termínů, technologií sítě a zařízení	11
2.3.2 Support	14
3 Návrh modelu aplikace	16
3.1 Typy problémů, příčiny, diagnóza a řešení	16
3.1.1 Chyby sítě a serverů	16
3.1.2 Chyby PC, HW a zařízení	17
3.1.3 Softwarové chyby	18
3.1.4 Ostatní požadavky	18
3.2 Helpdesk – doporučení, provoz a efektivita	18
3.2.1 Aplikování doporučení	19
3.3 Techniky návrhu – expertní systémy	20
3.3.1 Úvod do problematiky	20
3.4 Požadavky na aplikaci	21
3.5 Návrh flow chart s eskalací	22
4 Výběr implementačního prostředí	25
4.1 Studie vývojových platforem a prostředků	25
4.1.1 Java EE, NetBeans	25
4.1.2 PHP, „WAMP“	26

4.1.3	ASP.NET, Microsoft Sharepoint, Visual Studio	27
4.2	Porovnání s ohledem na cenu a použitelnost	27
4.3	Popis vybraných nástrojů a prostředí	28
4.3.1	Microsoft SharePoint 2010	28
4.3.2	ASP.NET	32
4.3.3	Microsoft Visual Studio 2010	33
4.4	Možnosti vývoje aplikace	34
4.4.1	Využití nástrojů	34
4.4.2	Základní řešení vývoje	34
5	Implementace systému a testování	35
5.1	Instalace a nastavení implementačního prostředí	36
5.1.1	Server	36
5.1.2	Nástroje pro vývoj	39
5.1.3	Klienti	39
5.2	Návrh systému	40
5.2.1	Uživatelské role	40
5.2.2	Model databáze	41
5.2.3	Funkce helpdesku	42
5.2.4	Postupy a logika řešení	44
5.3	Vytváření systému	45
5.3.1	Implementace	45
5.4	Řešení vytvořených prvků	49
5.4.1	Postup vývoje a aplikování jednotlivých komponent	50
5.5	Testování	54
5.5.1	Migrace na firemní server BNS	54
5.5.2	Migrace aplikace na nový server	55
5.5.3	Postup testování a dosažené výsledky	55
5.6	Další rozšíření a customizace produktu	56
5.6.1	Propojení s MS Exchange, Outlookem a Sociální sítě	56
5.6.2	Širší komerční využití	57
6	Závěr	57

Literatura	58
Příloha 1. Obrázky.....	62
Příloha 2. Zdrojové kódy.....	66
Příloha 3. Manuály	71
Příloha 4. DVD.....	72

1 Úvod

Ve firmě The Best Network Solution (dále jen BNS), zabývající se outsourcingem větších i menších firem pracuji jako technik přes 2 roky. Od zákazníků společnost dostává denně desítky požadavků k řešení. Jedná se jak o problémy koncových uživatelů, tak problémy na straně serverů a topologie celé spravované sítě. Většina těchto požadavků je nutné řešit ihned nebo v nejbližší možné době. Vzhledem k tomu, že se jedná o firmu s relativně nízkým počtem zaměstnanců v porovnání s počtem a velikostí spravovaných firem, je nutné řešené problémy co nejrychleji diagnostikovat a řešit.

K tomu momentálně slouží vytvoření a úprava formulářů (dopisování poznatků o problému) na základě doručené pošty na firemní mailovou adresu cílenou k podpoře. Toto řešení je jednoduché, nicméně ne moc efektivní. Je nutné udělat několik kroků, než se dostaneme k řešení samotného problému. A další problém může nastat při nedostatečné informovanosti nebo nedorozumění ve chvíli, kdy řešení převzou ostatní zaměstnanci firmy, případně když se jedná o špatně nebo nepřesně definovaný požadavek samotným klientem.

Proto jsem se s ředitelem firmy domluvil na tom, že pro tuto správu vytvořím komplexnější prostředí, které bude na základě přímo zadaného požadavku klientem automaticky diagnostikovat daný problém a vytvoří nám tak automaticky požadavek, který bude jasný a na základě obsažených informací bude možno jej okamžitě řešit. Důležitými faktory, které musím v práci zahrnout, je výsledná cena daného prostředí a použitelnost pro zákazníky různých firem, kde každá má svoje individuální prostředí, a může mít rozdílné problémy.

Dalším krokem důležitým faktorem bude customizace, v níž bude možné helpdesk nastavit přesně podle požadavků jistého zákazníka. Eventuálně zde vznikne možnost rozšířit produkt za hranice společnosti BNS a nabídnout ho i jiným firmám zabývajících se IT podporou.

Pro představu funkčnosti vytvořeného systému jako celku uvedu příklad. Každý zákazník spravované firmy bude mít k dispozici aplikaci umožňující zadání jeho problémů. Možné závady se budou rozdělovat do několika navržených kategorií, dále podle zadavatele a typu. Uživatel tedy popíše svůj problém, přičemž uvede i jeho bližší specifikace z nabízených možností (podrobnosti o sobě, HW, SW, obchod,...). Helpdesk na základě těchto a automaticky získaných (funkčnost prvků sítě) informací doplní k požadavku další potřebná data, která se zobrazí vyhodnocené technikovi jako ucelený požadavek. Tento výběr a upřesnění proběhne automaticky pomocí navrženého modelu expertního systému a troubleshootingu – definice postupů řešení, pro daný problém. Na tento by měl být schopen z uvedených dat okamžitě reagovat, bez dalších postupů. Do požadavku bude samozřejmě možné pro upřesnění doplňovat další informace. Dále by mělo být možné se pomocí dalších přidanych prvků helpdesku informovat o aktuálním stavu sítě a zařízeních.

Důležitým aspektem pak bude co možná nejvíce uživatelsky přívětivé prostředí, které tak předejde dalším nesrovnalostem při zadávání požadavků koncovým uživatelem. Nasazením nového systému do reálného provozu je tak možné významně ušetřit vynaložený čas a zvýšit celkovou přehlednost a efektivitu.

1.1 Popis jednotlivých kapitol

Firma, klienti a analýza požadavků

V první kapitole jsou přesněji popsány vybrané firmy, které jsou spravovány a upřesnění všech jejich možných požadavků. Jsou zde podrobnosti o spravovaných sítích firem, jejich infrastruktura, podrobnější popis využívaných technologií a popis řešení typů problémů, které mohou nastat.

Návrh modelu aplikace

Dalším krokem je vytvoření a upřesnění cílů aplikace a požadavky, které by měla splňovat. Dále je zde popis technik návrhu a vytvoření podrobného flow-chart model pro vyvíjený helpdesk s ohledem na podporované firmy a jejich požadavky. Podrobně je zde rozebrán i samotný systém helpdesk jako koncept, technik návrhu a to, jak by měl vypadat jeho provoz.

Výběr implementačního prostředí

V této kapitole se věnuji průzkumu několika vhodných vývojových prostředků a jejich podrobnější popis a porovnání. Následuje výběr toho nejpoužitelnějšího řešení pro naše cíle a jeho kompletní popis včetně možností využití, vhodných funkcí a rozbor postupu při vytváření.

Implementace systému a testování

Tato klíčová a nejpodstatnější část práce obsahuje postup vytváření samotné aplikace. Tedy postup celým vývojem a popis všech jeho jednotlivých částí. Dále je zde uvedeno i následné zasazení do reálného provozu a testování za běhu včetně nezbytných oprav, následných úprav a vylepšení.

Dosažené výsledky a možná vylepšení

Diskuze dosažených výsledků a přínosů celé této práce jako projektu pro společnost BNS. Obsahuje jednak popis dalších vhodných rozšíření systémů za hranice firmy, pro kterou bylo vyvíjeno, ale také možné úpravy a rozvoj celé aplikace komplexnějšího řešení pokrývajících více oblastí.

2 Firma, klienti a analýza požadavků

2.1 Seznámení s firmou BNS

Společnost The Best Network Solution, s.r.o. vznikla v roce 1997. Zabývá se IT/IS outsourcingem, nabízí zákazníkům kompletní řešení zahrnující externí správu počítačových sítí, včetně navrhování, projekce sítí včetně následné konektivity, server hosting, web hosting, kompletní servis výpočetní techniky, vývoje aplikací a informačních systémů a v neposlední řadě školení v IT oblasti a firemní prezentace.

Outsourcing znamená, že daná firma provozuje poskytované služby externě, což má své výhody zejména v technologické flexibilitě. Dále hlavně v nízkém investičním riziku pro zákazníka, lepšímu peněžnímu toku a přístupu k novějším technologiím [1].

BNS je Microsoft Gold Certified Partner, má tedy přístup k jejich nástrojům a plné podpoře. Co se tedy týká softwarového řešení, nabízí produkty zpravidla v rámci společnosti Microsoft. Pokud uvažujeme spravované koncové zařízení - PC, všechny mají nainstalován OS Windows, přesněji verze Windows XP nebo Windows 7, v případě serverů, jsou zde Windows Server, a to verze 2003 nebo 2008. Tuto skutečnost je nutné vzít v potaz při realizaci mého systému.

2.1.1 Struktura firmy

Společnost se skládá ze čtyř hlavních úseků a tím je obchodní a ekonomické oddělení, vývoj, oddělení techniků. Samostatným oddělením je pak správa serverů (pro případ poruch, oprav a změn). Každé z nich má na starosti jiný typ řešených požadavků, bude tedy nutné, aby se odpovídající požadavky dostali příslušnému oddělení a zahrnout do helpdesku i tuto možnost. Samozřejmě musí být připravena i možnost předávání požadavku mezi odděleními (např. po doplnění podrobností). Pro upřesnění, jaké problémy by mělo dané oddělení řešit:

Technický úsek – řešení HW a SW problémů koncových zařízení, serverů a sítí, havárie.

Servery – problémy týkající se změn a oprav na serverech.

Obchodní úsek – objednání, dodávka zboží nebo služeb.

Ekonomický úsek – reklamace, opravy.

Úsek vývoje – vývoj aplikací a webů.

2.2 Průzkum portfolia spravovaných firem

Společnost spravuje přibližně několik desítek firem, jsou mezi nimi soukromé i státní podniky, větší i méně významní klienti. Mezi podniky s nejrozsáhlejší správou patří například Delikomat, Super Zoo,

Top-Office, Celní jednatelství Zelinka, ZŠ, Proex 2000, G4S, Autocentrum Duba. Pro tento projekt i pro celkovou přehlednost vyberu tři vzorové firmy, které svým působením a strukturou kompletně zastoupí všechny z možných problémů, situací a požadavků, které mohou nastat. Ty jsou podrobně rozepsány v dalších částech této kapitoly.

Dalším aspektem je, že v každé z firem pokrývá správa určitou oblast a tak některé specifické aplikace, servery nebo části sítě může spravovat jiná společnost. V jednotlivých kapitolách o firmách uvedu důležité informace právě potřebné k sestavení modelu aplikace.

2.2.1 Druhy požadavků klientů

Veškeré požadavky, které je nutné řešit, lze pro přehlednost rozdělit do čtyř hlavních kategorií – výroby, back office, technické, obchod. Všechny požadavky každé spravované firmy lze vždy rozdělit do těchto kategorií. Vybrané firmy (viz níže) pro adekvátnost zahrnují každou z těchto kategorií. Pod jednotlivé kategorie spadají následující úseky a řešení:

<i>Výroba</i>	- pokladny, PC a zařízení ve výrobě, servisu, dílnách
<i>Technologické</i>	- vývoj aplikací, kabeláž, internet, konektivita
<i>Obchod</i>	- nákupy, nabídky, reklamace, školení
<i>Back Office</i>	- PC a zařízení v kancelářích (účetní, manažeři, sekretářky)

Podrobněji bude rozebírat a zabývat se jednotlivými druhy kapitola 2.3 Analýza požadavků. Tento krok je důležitý hlavně z hlediska budoucího rozlišení druhů požadavků od klientů a také návrhu odpovídajícího modelu aplikace.

2.2.2 Delikomat

Jedná se o jednu z významnějších a větších firem svého druhu působící v oblasti prodejních automatů se širokou sítí po celé ČR zaměstnávající přibližně 300 lidí. Tato společnost má centrálu v Brně, kde se nacházejí i hlavní servery a dalších 11 větších i menších poboček.

2.2.2.1 Popis a zapojení sítě

Jak je uvedeno na obr. 2.1, celá síť firmy je složena z šesti serverů v Brně a jednoho v Praze. Jsou rozděleny dle jména/použití a poskytují:

<i>Deli</i>	– řízení infrastruktury a domény, proxy, DNS, DHCP server, update services
<i>Fin</i>	– virtuální server, zálohy starého programu Navision
<i>Data</i>	– datový server, zálohy dat
<i>Mail</i>	– mailový server
<i>CRM</i>	– server CRM systému InTouch

NAVI, NAVI II – informační systém Navision a jeho terminál

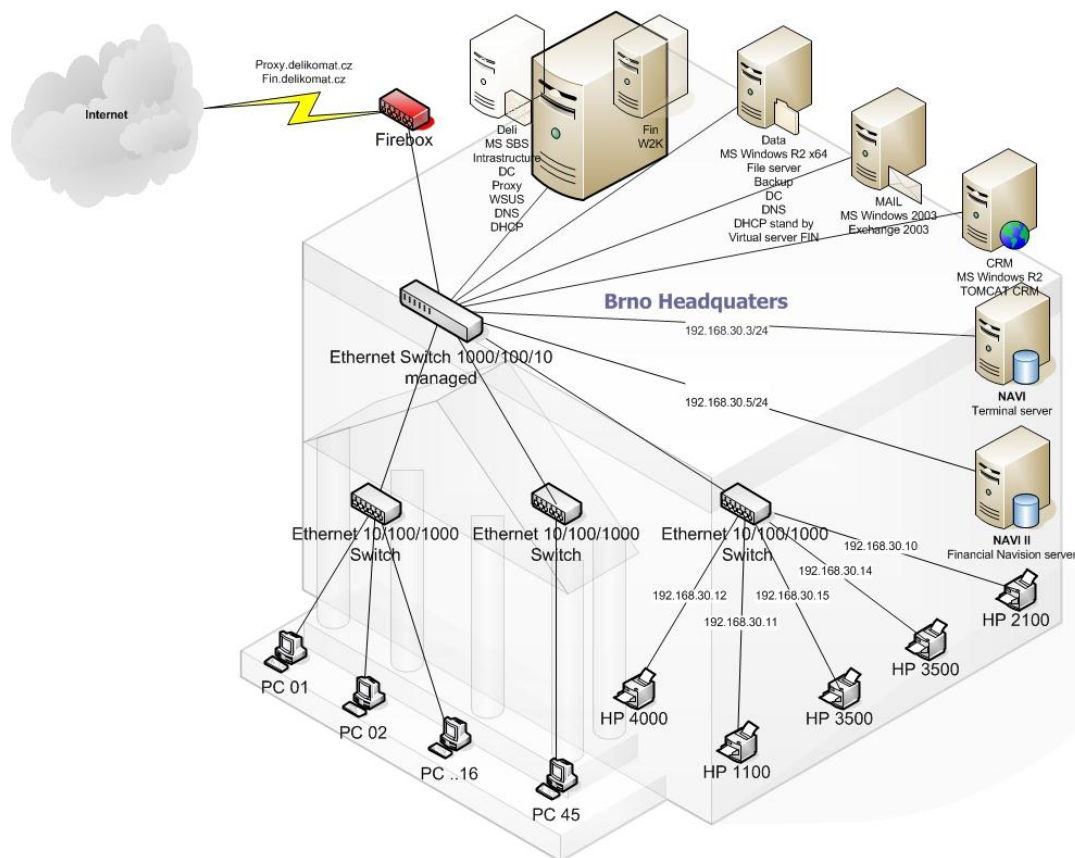
DeliPr – pražský DNS a datový server

Všechny servery a PC jsou propojeny a členy domény Delikomat, mají přístup do celé sítě a možnost přístupu na síťová úložiště. Z ostatní poboček se připojují k centrále pomocí VPN IPSec řešení (*WatchGuard Firebox* zařízení a klient). Všechny servery mají dané pevné IP adresy, stejně jako některé síťové tiskárny. Ostatním zařízením se přidělují automaticky pomocí DHCP. Vzdálená správa je prováděna vytvořením tunelu v programu *WatchGuard VPN* a následným připojením přes RDP v systému Windows na požadované zařízení.

2.2.2.2 Softwarové vybavení

Na serverech jsou provozovány operační systémy *MS Windows Server 2003* nebo *2008*. Na poštovním serveru je *MS Exchange 2003*. Dalšími pro firmu globálně důležitými aplikacemi jsou informační systém *Navision* pro zprávu podnikových zdrojů a *Intouch* pro správu zaměstnanců i zákazníků společnosti. Na PC koncových uživatelů jsou dále provozovány a pro bezproblémovou práci důležité *MS Office* včetně *MS Outlook* pro poštu, *Firebox klient* pro vzdálenou práci a připojení. Další instalovaný SW již není tak důležitý, jako výše zmíněné.

2.2.2.3 Topologie sítě



Obrázek 2.1: Schéma vnitřní sítě centrály firmy Delikomat v Brně.

2.2.3 Super Zoo

Jak již název napovídá, jde o firmu se sítí specializovaných maloobchodních prodejen zabývajících se prodejem chovatelských potřeb, krmiv a živých zvířat. Celkem má společnost 30 poboček po celé republice, z hlediska BNS tedy patří mezi nejrozšířenějšího partnera. Na každé pobočce se nachází vždy minimálně dvě PC, a to vždy jedno pokladní, které vyžaduje odlišnou správu a aplikace. Servery této firmy jsou umístěny přímo ve firmě BNS a všichni uživatelé se na ně přihlašují přes virtuální privátní síť. Řeší se zde hlavně problémy z technické oblasti a oblasti pokladen, u kterých je pokladní sw spravován další externí firmou Kelar s.r.o.

2.2.3.1 Popis síť

Na *obr. 2.2* lze vidět, že SuperZoo má celkem tři servery:

<i>Server</i>	- řízení infrastruktury, domény, proxy a dns server, mail server
<i>SAP</i>	- aplikace SAP
<i>Terminal</i>	- server pro vzdálené připojování uživatelů

Všechny PC v síti jsou členy domény SuperZoo a připojují se do ní pomocí VPN IPSec připojení, které je vytáčeno již přímo na přístupovém bodu k internetu - ADSL modemu. Veškerá PC včetně pokladen mají přidělovanou pevnou IP adresu pro snadnější správu. Pro správu je nutné vytvořit VPN tunel do sítě SuperZoo a dále se připojovat ke koncovým zařízením.

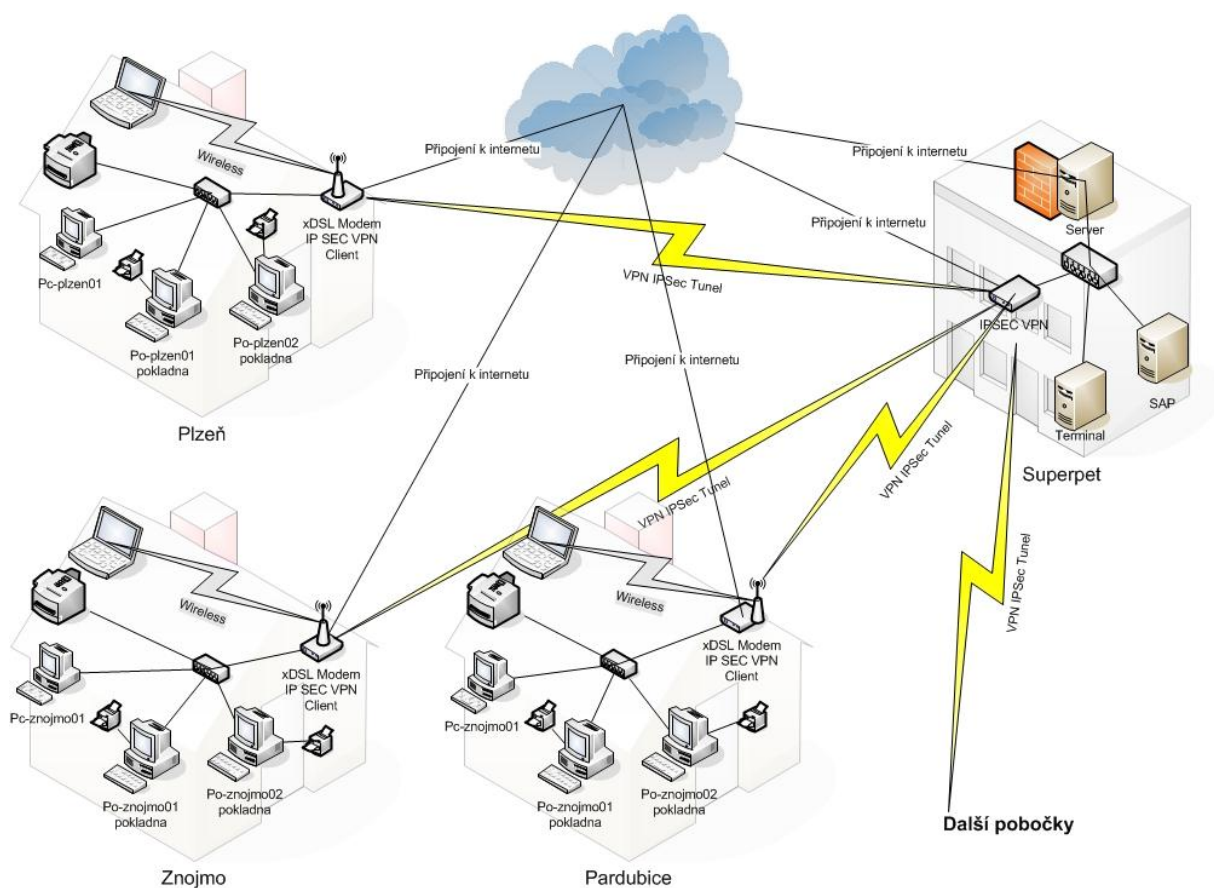
Do prodejen se postupně začínají připojovat IP kamery pro lokální i vzdálený monitoring. Ve všech prodejnách je také navíc zprovozněno wi-fi připojení k individuálnímu použití.

2.2.3.2 Softwarové vybavení

Na všech serverech firmy běží *OS Windows Server 2003* a pro mailové řešení je použit *MS Exchange 2003* ve spojení s *MS Outlook*. Důležitou aplikací je *SAP*, který zde slouží k podnikovému řízení, účetnictví a všichni uživatelé se k němu připojují přes vzdálený terminál (RDP).

Jako pokladní software je používán program *Winshop*, který je instalován v BNS, ale jeho samotná správa databáze a případné chyby jsou řešeny firmou Kelar. Veškeré požadavky na tuto aplikaci se tedy v BNS zpracují a předávají dále k řešení. Obdobný postup se uplatňuje i v rámci katalogu Plaček (databáze produktů), který je vyvíjen stejnojmennou firmou.

2.2.3.3 Topologie sítě



Obrázek 2.2: Síť firmy SuperZoo.

2.2.4 AxFlow

Je mezinárodní společností působící v celkem 20 evropských zemích. Specializuje se na aplikační know-how v rámci průmyslových segmentů a dopravních systémů kapalin čerpadel. BNS má na starosti českou i slovenskou pobočku této společnosti, nacházející se v Praze a Hlohovci. Protože se jedná o mezinárodní společnost se sídlem v Anglii, je zde nevyhnutelná komunikace se zahraničím, včetně předávání závažnějších problémů dále, případně postupu dle vedení.

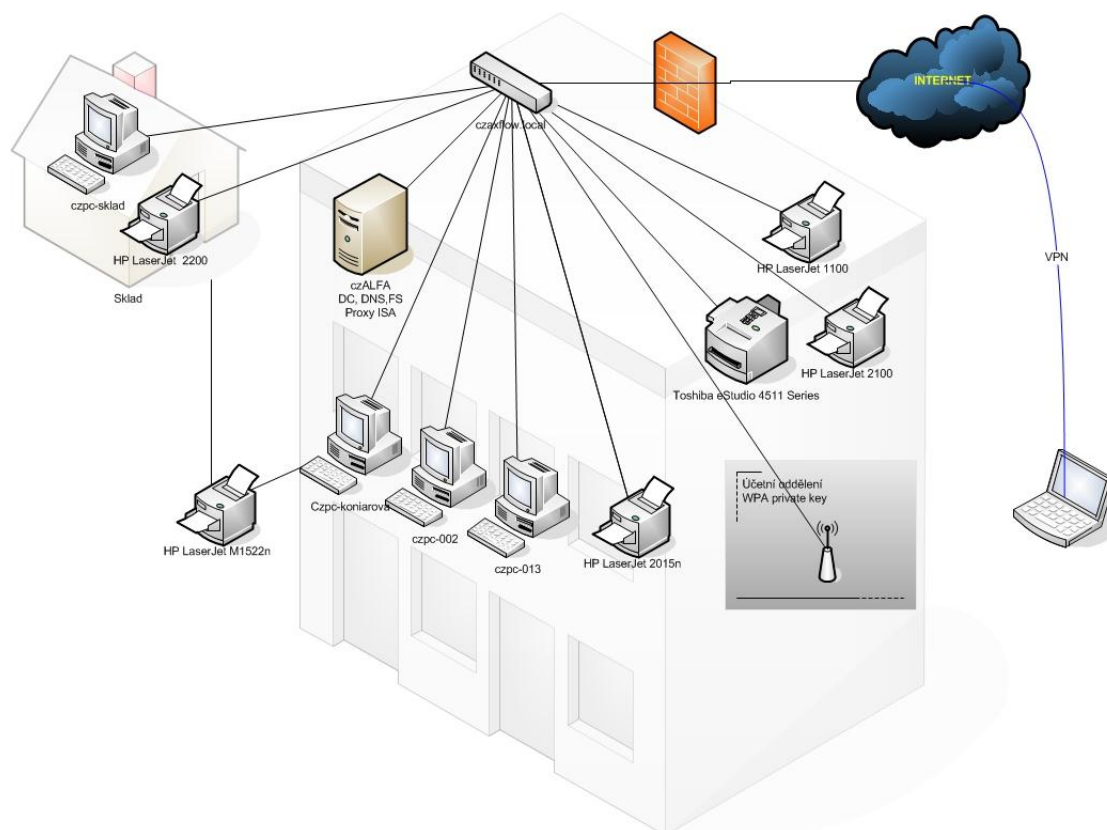
2.2.4.1 Popis sítě

Pro pobočku v ČR i SK je jeden společný server viz obr. 2.3. Je pojmenován *czAlfa* a funguje jako DNS a proxy server i jako file server. Všechna PC mají přiděleny pevné IP adresy. Obě pobočky jsou pomocí VPN propojeny s centrálou, kde je taktéž řešen poštovní server. Zařízení SK pobočky jsou rovněž propojeni se serverem *czAlfa* a mohou s ním pracovat. Pro vzdálenou správu je rovněž nutné nejdříve vytvořit VPN tunel pomocí daného programu pro připojení do sítě a následnou komunikaci v ní (RDP).

2.2.4.2 Softwarové vybavení

Na serveru je nainstalován OS *Windows Server 2003*. PC ve vnitřní síti a to v obou pobočkách jsou členem stejné domény Axflow. Celá vnitřní síť je pomocí programu *CheckPoint VPN* propojena s centrálou i ostatními pobočkami. Jako poštovní klient je zde využit *Lotus Notes*, s poštovním serverem v zahraničí.

2.2.4.3 Topologie sítě



Obrázek 2.3: Vnitřní síť centrály firmy AxFlow.

2.3 Analýza síťových technologií a požadavků

Při vytváření nových sítí a řešení pro zákazníky je ve firmě využito podobných nebo stejných postupů a technologií. Je tak později docíleno snadnější a přehlednější možnosti správy a kontroly nad všemi sítěmi a zařízeními. Dále jsou popsány využívané techniky potřebné pro organizaci sítě a její správu, které jsou pro moji práci nejdůležitější. Následuje výběr vzoru a analýza reálných příchozích požadavků z firemní pošty – supportu.

2.3.1 Popis využívaných termínů, technologií sítě a zařízení

Popis pro porozumění nejdůležitějších technik a technologií využitých v sítích vybraných i většiny ostatních firem spravovaných v BNS.

2.3.1.1 Termíny a technologie

Active directory/Domény

Všechny počítače a servery BNS jsou členy jisté domény, mají přidělen svůj unikátní název, některé i pevnou IP adresu vázanou k tomuto názvu.

Organizaci skupin počítačů a domén zajišťuje Active Directory, což je implementace adresářových služeb LDAP firmou Microsoft v prostředí systému Windows. Ta umožňuje nastavování síťových politik a provádět operace (instalace, aktualizace), poskytovat centrální služby pro autentizaci a autorizaci v celé její organizační struktuře, kterou jednoznačně definuje [2]. Jsou tak vytvořeny domény, což jsou skupiny počítačů sdílející společnou adresářovou databázi (mají jednoznačné označení, zásady zabezpečení a vytvářejí mezi sebou vztahy). Active directory je provázáno s DNS a využívá stejnou hierarchickou strukturu. Pracuje s daty, která jsou organizovány jako objekty, což jsou pojmenované skupiny atributů, které reprezentují síťové prostředky. Celé schéma active directory definuje objekty, které v něm mohou být uloženy a zároveň je seznamem definic, které určují druhy objektů a typy informací, které uchovávají.

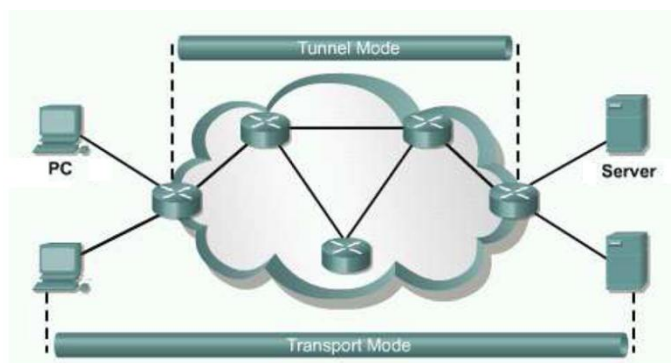
Využitý LDAP - Lightweight Directory Access Protocol – je aplikační protokol pro dotazování a modifikaci adresářových služeb a dat na adresářovém serveru nad TCP/IP, definuje jak si klienti a servery vyměňují informace o adresáři. Jednotlivé položky jsou na serveru ukládány formou záznamů a uspořádány do stromové struktury.

VPN IpSec

Ke spojení do vnitřních sítí firem se využívá VPN – virtuální privátní síť, které můžeme definovat jako privátní síťové služby přenášené přes veřejnou síťovou infrastrukturu. Jsou využívány jak pro správu, tak pro připojení poboček k centrálám firem.

Konkrétně je využíván protokol IPsec pracující na síťové vrstvě (vkládání speciální hlavičky pro doručení datových jednotek síťové vrstvy) a přístupových VPN sítí inicializovaných serverem. IPsec je bezpečnostní rozšíření IP protokolu založené na autentizaci a šifrování každého IP datagramu, poskytuje kontrolu přístupu, kontrolu integrity dat, autentizaci zdroje dat a důvěryhodnost

dat. Jakýkoliv přenos síťových aplikací je tedy zabezpečen. Využívá se při tom tunelového (zapouzdření původního IP datagramu a vytvoření nové hlavičky s vnější IP adresou) a transportního režimu (IP hlavička je nezměněna, chrání obsah transportní a vyšší vrstvy) IPsec technologie viz obr. 2.4.



Obrázek 2.4: Tunelový a transportní režim IPsec [3].

Lze použít asymetrické (s veřejným klíčem) i symetrické (s tajným klíčem) šifrování a pro přenos protokoly AH (autentizační hlavička) a ESP (zapouzdřený zabezpečený obsah) [3].

RDP

Pro připojení ke vzdáleným počítačům a serverům se používá RDP - Remote Desktop Protocol, systému Windows, který poskytuje uživateli práci v grafickém rozhraní na vzdáleném počítači, lze ho tak téměř kompletně spravovat. Všechny používané verze vzájemně kompatibilní (verze 6.1 a 7.0). RDP je rozšířením ITU-T (sektor telekomunikační standardizace) a T.128 (doporučení popisující komunikační a aplikační protokoly a služby) protokolů pro sdílení aplikací. Při spojení je využíváno TCP portu 3389 [4].

Proxy server

Působí jako zprostředkovatel mezi klientem a serverem při http komunikaci. Odděluje lokální počítačovou síť od vnější (internetu) [5]. Využívá se z mnoha důvodů a to hlavně to z hlediska bezpečnosti – pro cílový server je klientem proxy server a ne samotný klient; pro urychlení přístupu ke zdrojům - rychlejší přístup k webovému serveru pomocí ukládání často žádaných zdrojů do vyrovnávací paměti; pro aplikování politik přístupu ke službám; pro kontrolu datových přenosů.

DNS/DNS server

Domain Name System - jedná se o hierarchický systém doménových jmen realizován DNS serverem (nameserver). Má za úkol překlad IP adres zařízení v sítích na doménová jména a naopak. Slouží jako distribuovaná databáze síťových informací [6]. Protokol využívá port 53 v TCP i UDP komunikaci. DNS efektivně udržuje decentralizované databáze doménových jmen. DNS server poskytuje kromě překladu adres také další služby jako rozlišení aliasů počítačů, aliasy pro e-mail, rozdělení prostoru dat mezi více serverů, sdílení informací v rámci globálního prostoru jmen, delegování pravomocí za domény na jednotlivé subjekty.

DHCP server

Tam, kde nemají počítače přidělenou pevnou adresu, je využito DHCP. DHCP server automaticky přiděluje IP adresy na žádost hostů, kteří sdílejí společný adresový prostor a umožňuje tak jejich dynamickou konfiguraci. Pokud některý z nich vyšle zprávu o žádost přidělení IP na server, je mu odpovězeno nabídkami, host poté odešle potvrzení s výběrem. Po ukončení práce je adresa uvolněna a může být znovu použita [7].

MS Exchange/OWA

Jedná se o softwarový produkt společnosti Microsoft sloužící jako poštovní server, ke sdílení zdrojů a výměnu e-mailů. Je používán ve většině větších firem BNS. Mezi jeho vlastnosti

patří i poskytování řady dalších služeb, hlavně se jedná o správu kontaktů (adresářů), správa kalendáře, sdílení veřejných složek, možnost využití jako datového úložiště, přístup k systému pomocí mobilních zařízení a možnost přístupu k poštovním schránkám pomocí webového rozhraní – Outlook Web Access (OWA).

.NET Framework

Jde o základní komponentu .NET (soubor technologií v SW produktech) operačního prostředí pro programování a běh aplikací pro systémy Microsoft Windows. Jeho součástí je knihovna hotových komponent, s nimiž lze jako programátor pracovat. Má vlastní runtime prostředí.

2.3.1.2 Využívaná zařízení

Ve spravovaných firmách jsou jako přístupové body k internetu využívána tato zařízení. Jsou zároveň využity i jako VPN klienti s tím, že všechna připojená zařízení jsou také ve firemní síti.

WatchGuard / Firebox

Zařízení, které je využíváno ve firmě Delikomat, poskytující VPN pomocí IPsec a firewall, v infrastruktuře pracují také jako přístupový bod a síťová brána. Jedná se o kombinaci WatchGuard Firebox SSL Core VPN a Citrix Secure Access. Samotné zařízení umožňuje také vzdálenou správu včetně pokročilého nastavení pro přístup do sítě.

Nokia

Zařízení využitá ve firmě AxFlow, pracuje jako firewall, síťová brána a VPN pro připojení do zahraniční centrály firmy. Jako přístupový bod k internetu zde slouží Adsl modem.

Zyxel

Zařízení využívané ve všech pobočkách firmy SuperZoo (servery firmy umístěny v BNS) a některých pobočkách Delikomatu. Jedná se o Adsl modem/router, který zajišťuje firewall, připojení k internetu a vytáčení VPN do vzdálené firemní sítě. Router nabízí rozsáhlé možnosti pokročilého nastavení včetně jeho vzdálené správy.

2.3.2 Support

Současná podpora uživatelů funguje na bázi výběru příchozí pošty na adresu podpory (případně doplnění) do specifické veřejné složky na Exchange serveru a ruční doplnění o formulář s podrobnostmi (firma, kontakt, typ). Příklad aktuálního supportu můžeme vidět na *obr. 2.4*. Všechny jednotlivé požadavky tak můžeme rozdělit do kategorií, které jsou již upřesněny v předešlé kapitole druhů požadavků klientů.

Support (Hotline)		
Hledání: Support (Hotline)		
Uspořádkáno podle: Datum		Nejnovější nahoře
Dnes		
Ingr Lukáš	13:13	
PMBS - zápis z jednání		
Šimkovský Igor	12:24	
SABRA - starý sever OR		
IT Dispečink	12:10	
FW: Prosba		
Strnad Miroslav	12:02	
Dědeček počítač		
Liberec Kancelář	11:48	
prosba o připojení tiskárny		
Hanzlík Zbyněk	11:21	
ZŠ Pavlovská - účty žáků 6. tříd		
Příbram Kancelář	10:37	
problém		
Vašák Petr	9:36	
SuperPet - nefunkční klavesnice u pokladny		
Hanzlík Zbyněk	8:55	
BNS - kontrola rámečků		
Hanzlík Zbyněk	8:05	
PMBS - nový uživatel Janderková		
Včera		
Ján Marek	st 16:21	
Sabra - nedodělky o kterých víme		
Babšický Marek	st 13:18	
RE: export do Excelu z 6K		
Němeček Libor	st 13:11	
BNS - náhradní PC obchod		
Žondra Pavel	st 12:07	
intouch mapy - zobrazování		
Engel Vít	st 12:06	
Autoduba provedené práce		
Vopršal Jan	st 10:24	
ukládání scanu přímo do složek		
Praha-Štěrboholy kancelář	st 10:21	
FW:		
Úterý		
Šimkovský Igor	út 11:50	
Lenoxplus - Oldíci		

Často tak mohou vznikat nepřesné nebo špatné popisy problémů. Na *obrázku 2.5* je aktuální support požadavek (Outlook form) částečně automaticky vytvořený z příchozího mailu s možnostmi nastavení a upřesnění. Zde se ručně doplňují kontaktní informace, firma, problém, přílohy, přiřazení řešiteli, čas strávený řešením problému, případně další upřesnění. Lze v něm také jednoduše kontaktovat zadavatele nebo řešitele se všemi podstatnými informacemi.

Stěžejním bodem vývoje nové podpory je tedy samostatné řazení a automatické doplnění co možná nejvíce potřebných informací. Jak je vidět na seznamu požadavků, *obrázku 2.4* výše, jednotlivé požadavky jsou barevně označeny podle příslušnosti oddělení/zaměstnanci, je vidět také zadavatel a předmět (popis) problému. Podobné rozlišení a popis využijí i v připravovaném helpdesku, s tím, že na první pohled bude viditelných více podrobností, a nejdůležitějších poznatků.

Jedním ze současných problémů je například další dotazování zadavatele, kde dotazovaný odpoví buď opět na support a je tak nutné přepsání požadavků do jednoho, případně na individuální mail a musí se tedy informace přepisovat. Nová aplikace by měla být schopna jednodušší konverzace, tedy předání problému na zadavatele, případně

jeho informování o zadané otázce s možností jeho odpovědi a opětovné předání na řešitele. Bude tak vše uceleno a neztratí se žádná podrobnost.

^ Obr. 2.4:

Momentální podoba support a jednotlivé požadavky.

> Obr. 2.5:

Form požadavku ze supportu.

Od:	Opluštil Roman		
Předmět:	Zeisel - RAM		
Firma:	Zeisel	Problém:	RAM
Zadavatel:	Zeisel	Email:	rzeisel@zeisel-advokat.cz
		Telefon:	
<input type="checkbox"/> Globální problém <input type="checkbox"/> Internet funkční <input type="checkbox"/> Sdílení funkční <input type="checkbox"/> Pošta funkční			
Jméno technika:	Obchod	Typ zásahu:	Vzdálený
		Hodin práce:	0:00
Uložit		Nastavit zprávy	
Odesílatel Práce zahájena Informovat administrátora Práce dokončena		Zadavatel Práce zahájena Informovat zadavatele Práce dokončena	
Popis problému: Pan Zeisel chce rozšířit RAM na počítači PC-SEKRETARIAT. Dosavadní pracovní paměť, 256 MB (PC2100 DDR SDRAM) Na počítači je ještě jeden volný slot DDR SDRAM.			
Poznámky technika:			
Příloha co navrhuješ za ramky? dopiš to tam ať se to může předat obchodu. 2x512MB			

3 Návrh modelu aplikace

Cílem je vytvořit co nejpřesnější vývojový diagram výsledné aplikace. A to vzhledem k předem stanoveným požadavkům a doporučení, které jsou v této kapitole podrobněji popsány. Jednak jde o aplikování obecných doporučení a požadavků pro helpdesk, využití technik expertních systému, jednak postup dle zjištěných typů problému, které se v infrastruktuře vyskytují.

3.1 Typy problémů, příčiny, diagnóza a řešení

Tato část se věnuje popisu možných chyb, které mohou nastat. Ty je nutné co nejpřesněji diagnostikovat, a tak zjednodušit jejich řešení. Neexistuje přesný návod pro řešení každé chyby nebo problému, ale pomocí efektivních postupů, lze postupně zjistit příčinu a chybu tak odstranit. Pokud jde o vzdálenou správu, je základem zjištění podstatných informací od samotného uživatele, a to:

Q: Zda se jedná o problém v celé firmě (na jednom PC nebo u všech), jak dlouho trvá?

Q: Jestli jsou funkční síťové aplikace, jako pošta, síťové složky, internet?

Q: Byl již proveden restart PC, nebo postiženého zařízení/programu?

Q: Jaká se objevují chybová hlášení / projevy?

Veškeré problémy můžeme rozdělit do několika kategorií.

3.1.1 Chyby sítě a serverů

Jde o nejzávažnější a nejakutnější problémy, které je nutné řešit okamžitě, lze je označit také jako havárie. Bez konektivity k síti nebo serverům je řada využívaných aplikací nefunkční - nelze vykonávat požadovanou práci a zasažená firma tak přichází o prostředky a rychle strádá. Proto je nutná rychlá diagnostika a řešení. Příčin může být hned několik, a to:

nefunkční připojení k internetu ze strany poskytovatele; výpadek některého z prvků v síťové topologii; nefunkčnost aplikací na serveru; nefunkčnost samotného serveru; výpadek elektřiny.

Pro kontrolu konektivity využijí příkazy:

ping – příkaz pro ověření funkčnosti spojení mezi dvěma rozhraními, zasílá ICMP pakety hostiteli a čeká na jeho odpověď [8]. Nelze využít všude, na některých zařízeních je z důvodů bezpečnosti ping zakázán. Příklad: `ping jmeno.serveru.cz`

telnet - program pro připojení ke vzdálenému počítači pomocí telnet protokolu, který je založen na TCP komunikaci. Konkrétně použijí službu RDP, která běží na portu 3389, případně RPC (vzdálené volání procedur) na portu 135 [8]. Příklad: `telnet jmeno.serveru.cz 3389`

Servery a většina pro provoz podstatných zařízení ve firmách mají pevnou IP adresu a také doménové jméno. Pro diagnózu problému je nejdříve nutné vytočení VPN tunelu do firmy, pokud již toto nelze, a zároveň neodpovídají ani přístupové body, je zřejmě chyba na straně poskytovatele. Pokud je přístupové zařízení dostupné, ale nelze vytočit tunel do firmy, je problém na straně zařízení poskytující VPN připojení. Následuje kontrola připojení na server pomocí telnet, pokud neodpovídá, patrně je nefunkční. Dále je takto nutné zkontrolovat, zda všechny služby běží korektně. V opačném případě, zda jsou zařízení vůbec funkční/zapnuta pomocí uživatele. Tento postup je podobný pro kontrolu ostatních prvků v síti. Pokud nakonec nepomůže ani restart pro uživatele přístupného zařízení nebo serveru, je nutná prezenční návštěva a kontrola.

Poštovní server

Pokud jde o server, na němž běží pošta, provádí se více jednotlivých kontrol, protože se zde obecně může vyskytnout více dílčích problémů. Nejdříve se provede test dostupnosti pomocí příkazu telnet na port 3389 (zjištění funkčnosti/nefunkčnosti serveru) a kontrola přihlášení na OWA

`https://jmeno.serveru/exchange` (případně `https://jmeno.serveru/owa`, jde-li o Windows Server 2008), pokud stránku nelze zobrazit, je problém ve službách serveru MS Exchange. Dále v případě, že uživatel má problém s přijímáním nebo odesíláním pošty, provádí se kontrola pop3 příp. smtp služby následovně:

`telnet jmeno serveru pop3` – pokud proběhne a zobrazí se banner, je vše ok, v opačném případě je zastavená nebo nefunkční služba pop3 na serveru.

`telnet jmeno serveru smtp` – pokud se zobrazí banner s výpisem, je ok, v opačném případě je zastavena nebo nefunkční smtp služba na serveru.

3.1.2 Chyby PC, HW a zařízení

Jedná se o chyby na PC, pokladnách, tiskárnách a ostatních zařízení, které jsou využívány. Při zjišťování problému, zde hrají roli hlavně adekvátní otázky na zákazníka a při problémech s konektivitou podobný postup jako v předchozí kategorii (zjištění dostupnosti, nastavení IP adresy včetně ostatních relevantních parametrů).

Závažnost chyb je zde závislá vždy na konkrétním případě a zákaznickovy. Většinou však platí, že je vždy k dispozici záložní stanice (nebo prvek), která práci dočasně vynahradí, zatímco se pracuje na nápravě té původní. Co se týká pracovních stanic, pokud jsou funkční, většina chyb jde opravit vzdáleně, a to buď telefonátem, nebo připojením před RDP (pokud je k dispozici) a vlastním řešením. Pokud se zjistí na místě neopravitelná závada, je nutné zaslání nebo dovoz stanice na firmu, kde je problém řešen (záloha dat, přesun systému na novou stanici).

Ostatní zařízení jako tiskárny, síťové prvky atp. jsou předávány rovnou na reklamační oddělení (jsou-li v záruce), příp. na obchod pro novou objednávku.

3.1.3 Softwarové chyby

Řešení chyb vyskytujících se v aplikacích, případně nefunkčností jich samotných. Řada z nich, jako například poštovní klienti, pokladní software nebo účetnictví jsou závislé na aplikacích běžících na serveru, proto se často jedná o chybu na něm samotném případně o problém externí firmy. Zde je nejdůležitější zjištění, zda se jedná o problém jednoho uživatele, nebo v celé firmě. Na řadu programů již jsou vytvořeny postupy nápravy při výskytu „běžných“ chyb. Dalším druhem jsou problémy s programy, které spravují jiné externí firmy. Tyto jsou tedy předávány rovnou jim - pokud je zjištěno, že jde o chybu v samotném softwaru, ne chyby vázané na systém, zařízení nebo prostředky, které spravuje BNS. Jedná se zejména o pokladní a účetní software, případně firemní poštu.

3.1.4 Ostatní požadavky

Zahrnuje všechny ostatní požadavky, které nejsou nijak urgentní, jedná se o *obchodní požadavky*, *objednávky zboží*, *reklamace*, případně různá *sdělení a poptávky*. Tyto žádosti jsou předány na příslušné oddělení, které je diskutuje se zákazníkem a domluví se na nejschůdnějším řešení, případně vhodném termínu plnění.

3.2 Helpdesk – doporučení, provoz a efektivita

Z hlediska funkčnosti je cílem této služby poskytování pomoci nebo technické podpory koncovým uživatelům a firmám. Poskytuje uživatelům jednotné kontaktní místo, kde mohou získat pomoc při řešení jejich problémů. Můžeme jej rozdělit na interní nebo v našem případě externí, neboť zde poskytujeme outsourcing podporu externím zákazníkům a partnerům. Helpdesk jako služba může být poskytován různými způsoby, například telefonní linkou, emailem, aplikací, webem, většinou se však jedná o kombinaci těchto prvků. Další jeho funkcí může být sledování, shromažďování a vyhodnocování informací, které se následně využijí pro další vylepšování celé struktury v IT oblastech. Může dávat správcům možnost sledovat a dohlížet nad stavem požadavků a infrastruktury a zpřehledňuje tak veškerou činnost [9].

Jednotlivé požadavky, z obecného organizačního hlediska můžeme helpdesk rozdělit na několik úrovní. Na první, nejnižší, úrovni by měl být helpdesk schopen odpovědět na základní otázky a problémy, které jsou definovány bází znalostí (viz kapitola 3.3) nebo se jedná o běžné, nejčastěji kladené otázky. Pokud se problém nevyřeší zde, eskaluje na další úroveň, které jsou řešeny za pomoci definovaného troubleshootingu (postupy řešení problémů - později implementovaného ve vývojovém diagramu). Dále jsou zde zodpovědné osoby nebo skupiny (týmy), které mají na starosti řešení specifických problémů, a tak se i samotné požadavky rozdělují do jednotlivých kategorií. Dalším požadavkem na efektivitu je nutná přehlednost v kombinaci způsobů přenosu informací v případě

předávání a řešení požadavků (informování klienta nebo správce rozdílnými způsoby - telefonicky, emailem). Je tedy nutné všechny poznatky centralizovaně sepisovat k odpovídajícím problémům.

Mezi doporučení pro kvalitní helpdesk patří hlavně následující prvky [10]:

- *přístup přes webové rozhraní* – výhodnější jak z uživatelského hlediska (snadný přístup), tak kvůli „přenositelnosti“ systému
- *možnost konfigurace a eskalace* – nastavení parametrů, které se mohou pro každou firmu lišit
- *možnost nastavení uživatelských rolí* – přihlašování pod určitými uživatelskými rolemi (zaměstnanec, správce) a nastavení rozdílných práv a možností
- *přehledné informace o stavu požadavků* – zaznamenání každého problému i po dokončení daného požadavku, možnost pozdějšího vyhledání
- *jednoduše ovladatelné a přehledné uživatelské rozhraní* – možnost práce i pro méně zkušené uživatele, jasný přehled nad procesy a požadavky
- *možnost zasílání notifikací* – zasílání upozornění, případně změnu požadavku na emailovou adresu aktuálních uživatelů
- *dokumentace požadavků* – doplňování informací k požadavku, tisk/export, archiv, vyhledávání
- *zabezpečený provoz* – ochrana proti zneužití nebo napadení i nechtěným chybám

Všechny z těchto bodů plánuji do aplikace zahrnout.

3.2.1 Aplikování doporučení

Už při zadávání požadavku uživatelem je nutné, aby mu sama aplikace nabídla v závislosti na vyplněných hodnotách možné akce, které může sám uživatel provést pro vyřešení, bez kontaktu správce (těmito akcemi je myšlen například restartování PC, nebo některého síťového aktivního prvku, kontaktování providera atd.). Takto se, alespoň podle osobních zkušeností, významně sníží počet žádostí v systému čekající na vyřešení.

Dalším důležitým faktorem je obecná přehlednost systému. A to jak pro zákazníky, mezi nimiž se nacházejí i méně zkušené uživatele (nemělo by dojít k nedorozumění), tak pro správce, kdy je nutné, aby při prvním pohledu na požadavek bylo zjistitelných co nejvíce podrobností (jedná se o havárii, komu přísluší, atd.). To lze vyřešit přehledným a logickým uspořádáním jednotlivých prvků systému, barevným rozlišením a podobně.

Současným problémem je z velké části mimo jiné nepřesný popis požadavku při vedení incidentu. Proto bude přesná analýza problému vytvořena na základě jednoduchého popisu problému a výběru jednotlivých polí uživatelem při zadávání požadavku. To se musí odrazit také při samotném vedení této žádosti v systému, kde bude pro správce vše přehledně a jasně určeno.

3.3 Techniky návrhu – expertní systémy

Vyvíjená aplikace by měla být schopna při některých požadavcích (zejména v případě konektivity sítě a zařízení) rozhodovat/vyhodnocovat na základě vstupních nebo zjištěných údajů o možnosti řešení konkrétní situace, tedy provádět troubleshooting a postupovat podle vývojového diagramu. Při návrhu proto budu uvažovat některé z technik expertních systémů, které následně implementuji.

3.3.1 Úvod do problematiky

Expertní systémy mají svoje využití tam, kde se má systém samostatně rozhodovat a nalézt řešení, přičemž existují daná pravidla, podle kterých se řídí a vyhodnocují.

Definice: Jsou to systémy hledající řešení problému v rozsahu určitého souboru tvrzení nebo jistého seskupení znalostí, které byly formulovány experty pro danou specifickou oblast.

Expertní systémy se vyznačují následujícími charakteristickými rysy: *Oddělení znalostí, mechanismu jejich využívání, schopnost rozhodování za neurčitostí a schopnost vysvětlování* [11].

Pro mou implementaci je tento systém přímo ideální, a to hlavně z následujících důvodů.

- je schopen okamžitě vyhodnotit i složitější problémy, které by jinak trvaly mnohem déle.
- jsou dostupné trvalé a opakovatelné expertízy a s tím spojené snížené náklady na jejich provedení.
- uchovávání získaných odborných znalostí nezávisle na zaměstnancích.

Na druhou stranu zde hrozí nebezpečí selhání ve změněných podmínkách, to znamená, pokud implementuji známé mechanismy do nového prostředí (další spravovaná firma), je nutné počítat se změnou/úpravou pravidel a eskalací.

Struktura expertního systému obsahuje tyto základní složky:

Báze znalostí – obsahuje znalosti z určité domény a specifické znalosti o řešení problému. Tyto znalosti mohou mít různý charakter, v mém případě půjde o využití navrženého flow-chart diagramu, tedy rozhodovacího stromu a matematické logiky.

Báze dat – modul pro získávání znalostí a uchování potřebných dat k vyhodnocení.

Inferenční mechanismus – obsahuje obecné algoritmy schopné řešit problémy na základě zadaných faktů pomocí manipulace se znalostmi z báze znalostí. Využiji mechanismus založený na Dedukci – logickém usuzování, při němž závěry musejí vyplývat z předpokladů.

I/O rozhraní – uživatelské, vývojové, vazby na jiné systémy, tedy vlastní helpdesk.

Proces tvorby expertních systémů – podle násl. činností budu postupovat při vývoji aplikace:

Výběr hardwaru a softwaru – specifikace požadavků pro běh aplikace.

Návrh uživatelského rozhraní – návrh prostředí, aplikace na webu nebo v systému.

Akvizice znalostí – získání znalostí a reprezentace jak od uživatele, tak pomocí automatické kontroly a diagnostiky.

Implementace – vývoj a tvorba ve vybraném prostředí.

Validace a verifikace – kontrola, zda diagnostika probíhá správně, nasazení do provozu

Pro smysluplné využití expertního systému by měly být splněny následující dvě podmínky:

1. Musí se jednat o problém složitý rozsahem nebo neurčitostí vztahů, pro nějž exaktní metoda řešení buď není k dispozici, nebo není schopna poskytnout řešení v požadované době.
2. Efekty plynoucí z použití expertního systému musejí převyšovat vynaložené náklady. To znamená, že by mělo jít o problém s opakovanou potřebou řešení a značnými finančními dopady, pro nějž lidští experti jsou drazí nebo omezeně dostupní.

Vyhodnocování bude probíhat na základě vývojového diagramu, podle kterého bude systém sestaven, obsahující veškerá možná řešení a na základě dostupnosti/funkčnosti prvků v síti. Dostupnost se bude automaticky testovat na základě druhu požadavku anebo rozhodování dle flow-chart.

Příklady uplatnění expertního systému v reálném prostředí:

- pokud není dostupný server → vyhodnot' jako havárii a proved' další body řešení.
- pokud byla hlášena nefunkční pošta → proveř přístup na poštovní server, zkontroluj služby pop3, smtp, webmail a tak zjistí, co není funkční.
- pokud je hlášena závada, ale nejde o práci v paušálu → předej obchodu na cenění.

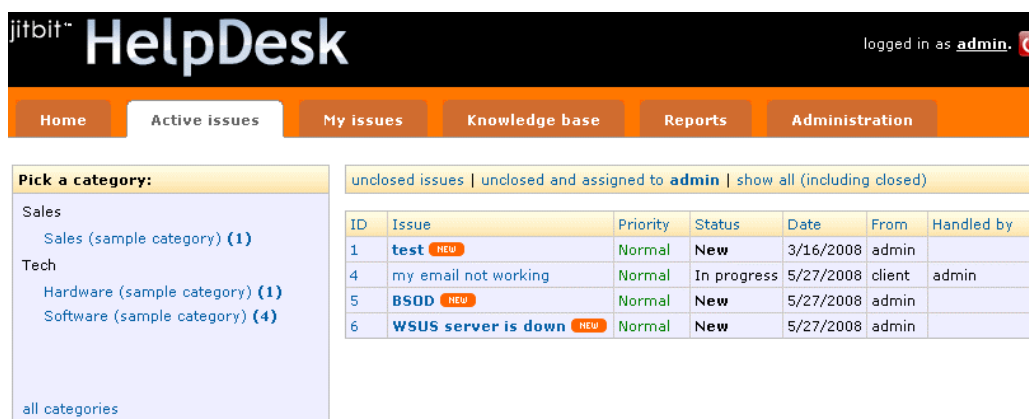
3.4 Požadavky na aplikaci

Výsledná helpdesk aplikace by měla mimo jiné splňovat alespoň minimální požadavky na provoz, správu a efektivitu uvedené v kapitole 3.2.

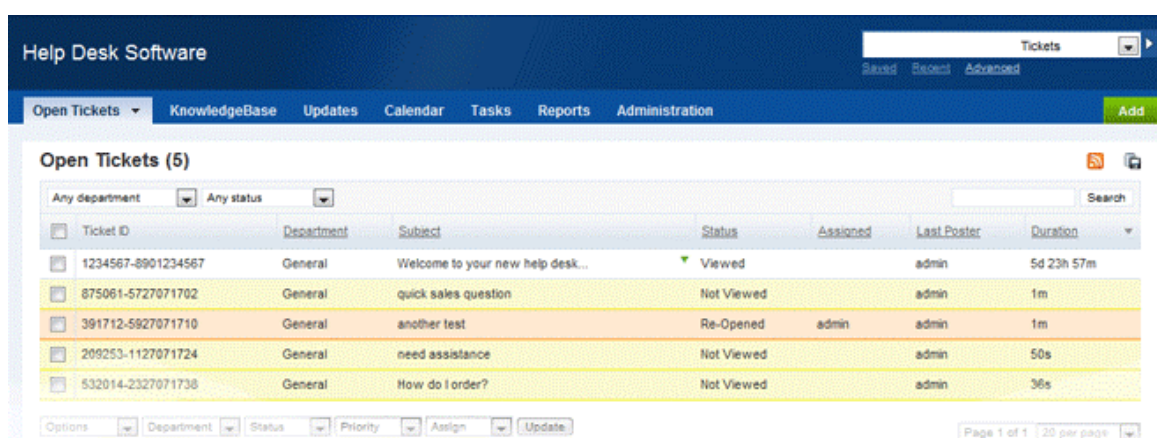
Co se týká specifikací pro samotný běh, ty vycházejí hlavně z možností a nastavení koncových zařízení, na kterých bude helpdesk provozován. Pokud tedy půjde o systémovou aplikaci, je důležitý bezproblémový běh pod OS Windows XP, Windows 7 a její minimální náročnost na výpočetní výkon. V případě webové aplikace je nutná plná kompatibilita s prohlížečem Internet Explorer 6,7, minimální zatížení síťového provozu a výpočetního výkonu samotného zařízení. V obou případech bude helpdesk testován a spuštěn na serveru se systémem Windows Server 2008.

Neméně důležitým požadavkem je uživatelsky přívětivé a nenáročné prostředí, ve kterém se půjde snadno orientovat. Obě strany uživatelů, tedy klienti i technici musí mít především jasný přehled o vedení a stavu incidentů. Na straně klientů je nejdůležitější vytvoření adekvátně rozvrženého formuláře pro zadání požadavku tak, aby se zamezilo nedorozuměním, která v tuto chvíli často vznikají. V případě techniků je nejdůležitější okamžitý přehled o stavu požadavků a co nejvíce podrobných informací týkajících se daného problému. Dále také dopisování dalších poznatků do požadavku a možnost rychlé komunikace/dotazování se zadávajícím.

Dalším možným krokem je propojení helpdesku s Microsoft Exchange Serverem, emailovým klientem Outlook a synchronizace jednotlivých požadavků i v těchto programech. Zde je ovšem zatím brzo na to posoudit, jak moc by bylo v našem případě toto řešení prospěšné a efektivní. Na obrázcích 3.1 a 3.2 jsou zobrazeny příklady vzhledů aplikací pro správu od různých vývojářů.



Obrázek 3.1: Aplikační Jitbit helpdesk [28].

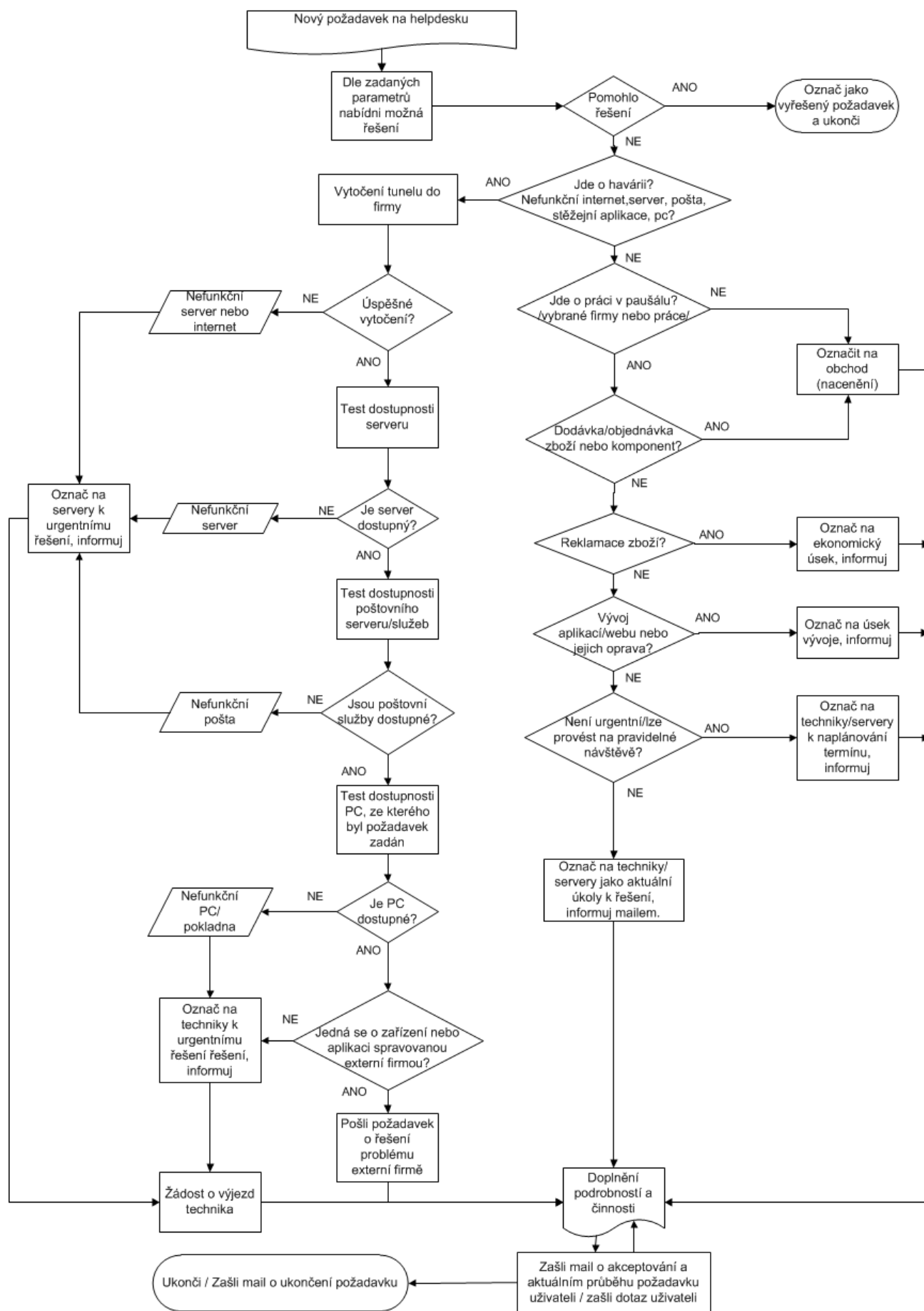


Obrázek 3.2: Webový helpdesk firmy Active Campaign[28].

3.5 Návrh flow chart s eskalací

Jedním z nejdůležitějších kroků před samotným vývojem je vytvoření diagramu prezentující procesy a algoritmy v samotné aplikaci. Důležitým aspektem je zakomponování eskalace pro jednotlivé firmy, kde každá může využívat rozdílná řešení, programy a mít odlišné problémy. Nicméně podle navrženého diagramu, by měl být každý požadavek správným a universálním způsobem zařazen a krok za krokem řešen.

Podle diagramu by měla být aplikace samostatně schopna řešit část problémů, a to buď rozhodováním a přiřazením do určité kategorie, nebo otestováním funkčnosti (dostupnosti zařízení, kontrola připojení) pomocí řešení uvedených mimo jiné v kapitole 3.1.



Obrázek 3.3: Flow-chart systému helpdesk.

Jak je uvedeno ve vývojovém diagramu (*obr. 3.3*), prvním krokem je samostatná diagnóza podle zadaných parametrů uživatelem a následné nabídnutí možných řešení, ještě před vytvořením nové žádosti, podobě zobrazovaných typů na řešení. Pokud je takto problém vyřešen, požadavek se pouze uloží jako vyřízený. V opačném případě je rozhodnuto podle závažnosti problému (zadání uživatelem), zda se jedná o havárii (a je nutné okamžité řešení) nebo běžný problém, který nikterak neomezuje provoz firmy nebo jednotlivce.

Pokud jde o nezávažný problém je nutné nejdříve podle firmy určit, zda má zaplacený paušál na naše služby, pokud ne, je předáno na obchodní oddělení na ocenění požadované práce. Na obchodní oddělení je předáno i v případě objednání nebo dodání zboží a komplement. Dále pokud jde o reklamaci - předáno ekonomickému oddělení, pokud jde o vývoj nebo opravu aplikací – předáno na vývoj/programátorům, a nakonec pokud jde o technické problémy je předáno technikům, kteří se podle závažnosti domluví s uživatelem na možnosti řešení ihned, nebo je zařazeno do seznamu úkolů firmy a řeší se na další naplánované pravidelné návštěvě (toto rozdělení probíhá také automaticky na základě vstupního zadání problému uživatelem). Následně je proveden pro každý problém všemi úseky případné doplněná provedená činnost (požadovaný čas) a doplnění důležitých informací. O každém přiděleném problému je příslušný úsek automaticky informován mailem.

Pokud se jedná o urgentní problém – havárii, je nejdříve otestována funkčnost vzdáleného připojení do dané firmy. Dále pak funkčnost serverů, případně služeb a koncových zařízení, které jdou vzdáleně otestovat. Výsledky jsou zaznamenány a označeny na příslušné techniky (servery, PC), případně na externí firmu, která dané služby poskytuje. Pokud nelze problém řešit vzdáleně, je přidělena i žádost o výjezd technika na požadované pracoviště. Nakonec jsou opět doplněny zbylé získané informace a podrobnosti, včetně času stráveného nad problémem.

Posledním krokem po veškerém vyhodnocení je informování uživatele o stavu a průběhu požadavku automaticky mailem (případně telefonem), případně předání zpět zadavateli s doplňující otázkou. V průběhu řešení následují další doplnění získaných informací a činnosti do požadavku. Požadavek je ukončen/vyřízen přechodem do koncového bodu pokud je žádost vyřízena.

4 Výběr implementačního prostředí

Při výběru je nutné zaměřit se a projít vývojové nástroje, které se nejvíce pro zadaný projekt hodí. Jde tedy o nástroje schopné pracovat s databázemi, webovým rozhraním a vývojem na serverech, kde bude také výsledná aplikace běžet. Dalším důležitým aspektem, který je třeba brát v úvahu je výsledná cena aplikace, tedy i cena samotných nástrojů a výdajů, které na ně budou vynaloženy.

Pro zvolení tedy vyberu několik nejvíce vhodných kandidátů a provedu jejich srovnání, z hlediska funkčnosti i jejich ceny.

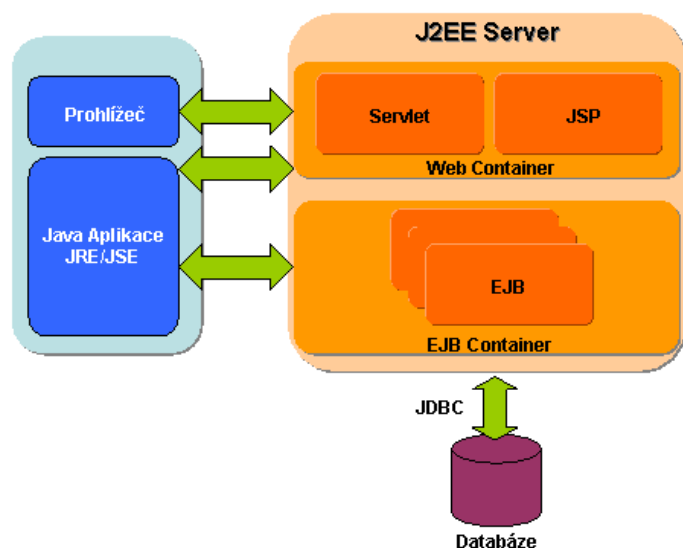
4.1 Studie vývojových platforem a prostředků

Seznámení a popis ze tří nejvhodněji použitelných prostředí pro budoucí vývoj mojí aplikace.

4.1.1 Java EE, NetBeans

Java EE (do verze 1.4 známá jako J2EE) tedy Java Enterprise Edition (aktuální verze Java EE 6) je široce využitelná platforma založená na jazyce Java vyvinutého společností Sun Microsystems, momentálně pod společností Oracle. Tato verze balíku technologií je určena hlavně pro serverové aplikace a využívá se zejména jako nadstavba pro rozsáhlé vícevrstvé podnikové aplikace. Oproti standartní verzi má k dispozici více knihoven obohacující funkcionalitu a je založena na modulárních komponentech běžících na aplikačním serveru.

Na diagramu níže (*obr. 4.1*) můžeme vidět složení J2EE z aplikačních komponent (klientské aplikace, applety, servlety, JSP, EJB) a kontejnerů poskytující runtime pro manažery jednotlivých systémových zdrojů. Ti jsou určeni pro správu systémových zdrojů. Architektura J2EE je základem integrace všech typů aplikací a určuje jak je aplikace vyvíjet.



Obrázek 4.1: Diagram popisující J2EE platformu [16].

Základem celé architektury je verze Java SE tedy Standart edition. Jedná se o základní funkční API a může obsahovat i vývojové knihovny, tzv. JDK – Java Development Kit, na kterých lze spouštět další aplikační služby. Hlavní výhodou Java EE je už výše zmíněné serverové části vývoje. Výhoda je zejména v samotné API, které je navrženo pro snazší a rychlejší vývoj, takže se lze soustředit primárně na aplikační úlohy. Mezi primární součásti celého balíku patří tyto:

- *JSP (Java Server Pages)* – určeno pro generování dynamického obsahu pro webové prohlížeče a mobilní zařízení (generování HTML stránek s výsledky dotazů).
- *JSF (Java Server Faces)* – komponenty generující uživatelské rozhraní na straně serveru, bez ohledu na výsledný protokol (snazší a rychlejší vývoj aplikací)
- *Servlety* – jsou základem JSP, které jsou po kompilaci generovány do servletu. Užívají se na optimalizované zpracování dotazů. Ve větších systémech slouží primárně rozdělování dotazů a dat na EJB v aplikačním serveru.
- *EJB (Enterprise JavaBeans)* – jedná se o hlavní součást Java EE. Jsou 3 typy - „session beans“ starající se o aplikační logiku, „entity beans“ pro práci s perzistentními daty společně s „message-driven beans“ pro asynchronní aplikace.
- *JDBC* – API umožňující unifikovaný přístup k libovolné RDBMS (relační systém řízení báze dat) s odpovídajícími ovladači.

Toto jsou hlavní technologie. Kromě nich je obsažena ještě řada dalších, které jsou spravovány aplikačním serverem. Tyto poskytují například přístupování k adresářovým službám (LDAP), sady tříd pro modelování poštovního systému, přístup a monitorování aplikací a aplikačních objektů, atd.

Vývojových nástrojů je k dispozici hned několik, každý z nich má své klady i zápory, ale dá se říci, že v nich můžeme vyvíjet bez větších omezení. Mezi ty nejznámější patří například Eclipse, JetBrains IntelliJ, Oracle JDeveloper. Vybral jsem jeden z nejrozšířenějších, kterým je NetBeans IDE. Ten představuje open source nástroj a vývojové prostředí postavené na platformě Java. Je tedy bezplatné a volně distribuovatelné i v komerčním sektoru. Stejně jako Java, pochází od vývojářů Sun Microsystems. Primárně je určeno pro vývoj v jazyce Java (podporuje ale i další). Je dostupná také vývojová platforma, což umožňuje rozšířit aplikaci pomocí modulů pro vytváření rozsáhlých aplikací. Aplikaci lze překonfigurovat dle vlastních potřeb pro určité úkoly a to volbou edice, pluginů nebo platformem.

4.1.2 PHP, „WAMP“

WAMP (případně WIMP) je zkratka (tvořená z iniciálů produktů Windows, Apache/IIS, MySQL, PHP/Pearl/Python) označující sadu softwarových produktů, používaných jako platforma pro implementaci dynamických webových stránek. Je to jedna z alternativ pro daný operační systém vycházejících z LAMP, což je stejná sw sada běžící pod OS Linux [17]. Protože jsou jednotlivé technologie, ze kterých je balík složen vyvíjeny samostatně, lze je nahradit různými alternativami.

Kromě nutné licence, která je potřebná samotný využitý OS se jedná o sadu open source software, která je zdarma. Také díky snadnému psaní kódu, rychlému nasazení a možnosti lokálního vývoje se stalo toto řešení velice oblíbeným. I když se jednotlivé technologie vyvíjeli samostatně, možnosti vývoje a sady nástrojů jsou sdíleny a byly vyvinuty v úzké spolupráci. Pokud jde o jednotlivé produkty:

- *Apache http server* – jedná se o open source webový server. Tento software hrál klíčovou roli při vývoji WWW (v roce 2009 se stal prvním web serverem, který překročil hranici 100 milionů webových stránek). Nyní je dostupný pro použití ve většině OS. V základu podporuje řadu funkcí, které jsou rozšiřitelné pomocí implementování modulů (např. PHP modul).
- *MySQL* – multiplatformní databázový systém. Komunikace zde tedy probíhá pomocí jazyka SQL (případně s jeho rozšířeními). Systém je oblíben hlavně díky výkonu a snadné implementovatelnosti.
- *PHP* – reflexní (lze pozorovat a měnit vlastní strukturu a chování za běhu) programovací jazyk navržený pro vytváření dynamických webů. Je využíván zejména pro aplikace na straně serveru.
- *Pearl, Python* - interpretované dynamické programovací jazyky, alternativy podobně využitelné jako PHP.

4.1.3 ASP.NET, Microsoft Sharepoint, Visual Studio

Soubor navzájem kompatibilních vývojových platforem a prostředků od společnosti Microsoft. Tedy aplikační webový framework, rodina softwarových produktů a vývojové prostředí. Jejich podrobnému popisu se detailně věnuje následující kapitola 4.3.

Co se týká komerční stránky a potřebných licencí, SharePoint Foundation je poskytován k licenci Windows Serveru zdarma jako jeho nadstavba pomocí updatu. Dále pak SharePoint Designer potřebný k samotné tvorbě je volně ke stažení a k dispozici bezplatně při nekomerčním využití [18].

4.2 Porovnání s ohledem na cenu a použitelnost

Při výběru vhodného prostředí je v první řadě nejdůležitějším aspektem možnost běhu a plná kompatibilita s využívaným Windows Serverem 2008 (případně 2003), IIS, SQL Serverem a Active Directory. Výhodou přitom může být i práce s Exchange serverem.

Pokud budeme porovnávat použitelnost, nejvhodněji se jevícím produktem bude vývoj pomocí Microsoft SharePoint, kde získáme plnou kompatibilitu se všemi potřebnými službami a zároveň jednodušší celkové propojení produktů. Navíc máme stále možnost vytvořený program a jeho zdrojový kód vyvíjet a upravovat v libovolném podporujícím editoru a obohacovat tak výsledný

produkt. Z hlediska cenové relace, nejsou zde žádné výdaje, protože postačuje již vlastněná licence na produkt Windows Server.

Následujícím nástrojem, vhodným pro naše účely, je pak Java EE. Ta je přístupná také zdarma, ovšem z hlediska použitelnosti v našem prostředí oproti předchozímu produktu strádá. Jistě je možné ve výsledku vytvořit vše, co je požadováno, ale zřejmě s vyšším úsilím a časovými nároky.

4.3 Popis vybraných nástrojů a prostředí

K vytvoření samotné aplikace jsem se rozhodl využít kombinace prostředí MS SharePoint v kombinaci s podporovanou technologií webového aplikačního rozhraní ASP.NET a vývojářským nástrojem Visual Studio. Všechny tyto vývojové prostředky včetně jejich částí a možnosti použití ve vývojovém prostředí dále podrobněji popíši.

4.3.1 Microsoft SharePoint 2010

Jedná se o rodinu softwarových produktů společnosti Microsoft a jednotnou platformu pro spolupráci a informování v rámci organizace. Na této platformě lze budovat jednotlivé oblasti - vytvářet firemních portály, možnosti projektové spolupráce, sdílet informace, soubory a dokumenty, vytvářet web. Mezi tyto produkty patří [12]:

4.3.1.1 SharePoint Foundation

Do verze 3.0 známé jako Windows Sharepoint Services - WSS. Je volně přístupná nadstavba (pomocí updatu) pouze pro Windows Server 2003 a 2008 poskytující webový portál s běžně potřebnými funkcemi (je také součástí .NET Frameworku viz kap. 2.3.1.1). Zároveň tento balík zahrnuje kompletní objektový model a API rozhraní podporující technologii SharePoint (pro vývoj v něm je tato nadstavba nutností). Tato služba je základem veškerých SharePoint technologií a produktů, má veřejný a zdokumentovaný objektový model svých datových úložišť a funkcností, takže je snadné ji s využitím technologie .NET Framework dále rozšiřovat (viz obr. 4.1). SharePoint Foundation zahrnuje několik hlavních komponent:

- *Základní rámec* včetně objektového modelu, uchovávání obsahu a konfiguraci databáze SQL serveru, ovládací prvky ASP.NET pro presentaci obsahu
- *Správu webu* pro řízení tzv. „webových farem“ (viz dále), které jsou složeny z jednoho nebo více serverů s výskytem na jedné nebo více webech.
- *Kolekce a šablony* stránek pro rychlé vytvoření nebo úpravu vlastních stránek
- *Editovací nástroje* k modifikování struktury webu, úpravu vzhledu, vytváření a úpravu pracovního prostoru a objektů
- *Indexer* pro indexování databáze
- *Konfigurační nástroj* pro veškerá nastavení a konfiguraci

SharePoint spravuje obsah pomocí **IIS** - *Internetové Informační Služby*, což je webový server aplikace s řadou rozšiřujících modulů a funkcí od společnosti Microsoft pro použití se systémem Windows (je součástí systému Windows Server). Jako úložiště svých konfiguračních dat a dat obsahu využívá Microsoft SQL Server (případně interní databázi systému Windows). Pro každou webovou aplikaci se vytváří zvláštní databáze. K oprávnění skupin a uživatelů využívá podobný LDAP model jako Windows a může tak spolupracovat s Active Directory. Z hlediska licence, pokud nejde o vývoj komerčních portálů, je tato služba poskytována zdarma.

Fyzická struktura

Z tohoto hlediska je služba rozdělena do tří složek. První z nich jsou databázové stroje, kde jsou uloženy konfigurace a obsahy služeb. Tyto mohou být rozděleny do několika částí, a to podle komponent.

Další částí jsou podpůrné služby, provádějící úkoly na pozadí. Jsou to služby, které spouští plánované úkoly, služby pro zpracování dat, vyhledávací služby atp. Poslední částí jsou webové servery, na kterých běží webové aplikace. Jedná se o grafický interface sloužící zejména uživatelům pro přístup do služeb.

Logická struktura

Základem každé instalace SharePoint Foundation je tzv. **farma** [13]. Jedná se o logické uspořádání všech obsažených komponent v instalaci. Běží v ní všechny výše zmíněné služby. Farmu můžeme rozšiřovat o další podpůrné komponenty. Jsou jimi balíčky řešení a doplňkové funkce (webové aplikace, webové části, ovládací prvky) samotného WSS.

Součástí farmy je vždy tzv. **webová aplikace**, což je prostor uživatelského rozhraní. Každá webová aplikace má přidělenou svojí identitu (účet v Active Directory) a vyhrazený paměťový prostor. Data jsou umísťována na dvou úložištích, jednak v souborovém systému webového serveru, kam se ukládají statická data (funkční součásti aplikace) – prezentační vrstva, a jednak v databázovém serveru, kde je umístěn obsah (data, dokumenty, stránky). V rámci webových aplikací můžeme vytvářet také kolekce webů, což je logické uskupení webu - samostatné úložné prostory, ve kterých je umístěn obsah.

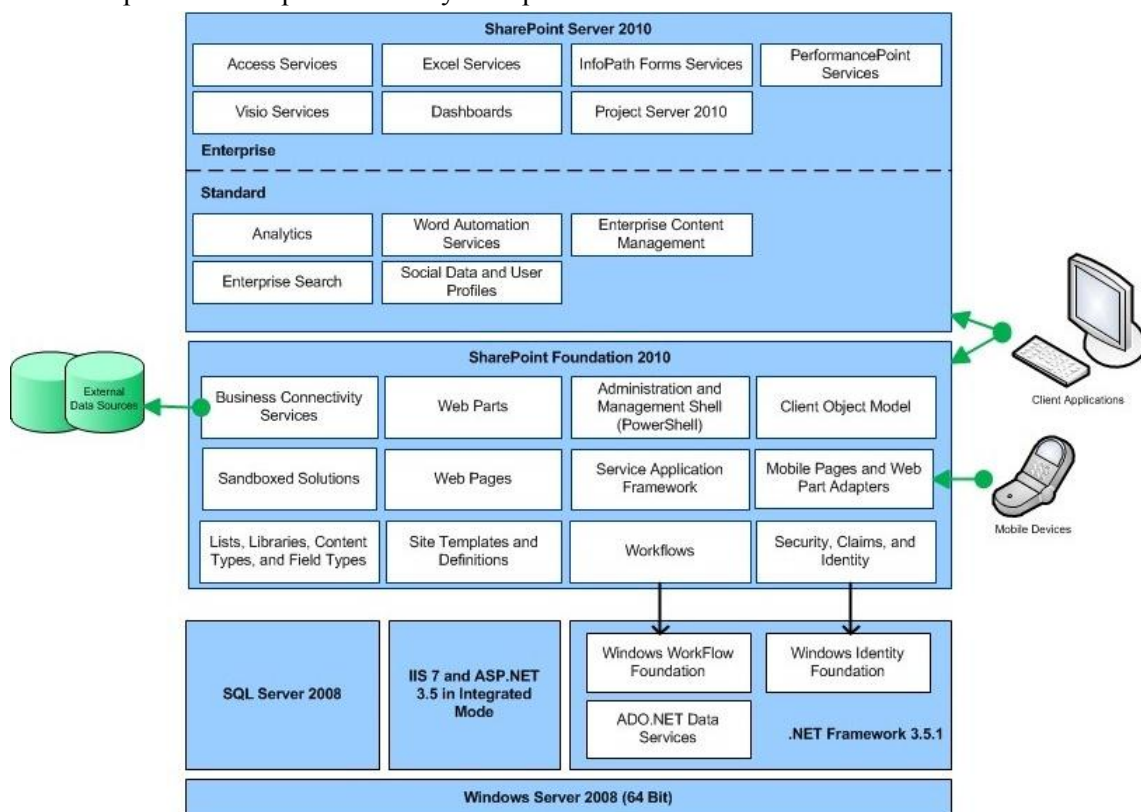
4.3.1.2 SharePoint Server

Jak je zobrazeno na *obr. 4.1* hierarchie vývojové platformy, SharePoint Server poskytuje další rozšířené služby nad rámec SharePoint Foundation (jehož služby jsou v něm také zahrnuty). SS obsahuje navíc řadu nástrojů [14]. Mezi ty nejpodstatnější patří:

- *Vývoj portálů*, určený pro sdílení informací s ostatními a přizpůsobení uživatelských zkušeností a obsahu podnikových webových stránek na základě profilu uživatele.

- *Vyhledávání* je rychlejší a efektivnější vyhledávání mezi lidmi, znalostmi a obsahem.
- *Management obsahu* je určený pro produktivnější vytváření a spravování dokumentů, záznamů a webového obsahu.
- *Byznysové procesy a formuláře* lze vytváření pracovních postupů a elektronických formulářů pro automatizaci a zefektivnění obchodních procesů.
- *Byznysová inteligence* umožňuje snadnější přístup k důležitým obchodním informacím, analyzovat a prohlížet data a publikovat zprávy.
- *Access, Excel a Visio services* umožňují propojení a operace s MS Excel, MS Access a MS Visio.
- *Webová analýza* zlepšuje výkon serverových farem pomocí plánování serverových zdrojů.
- *Zlepšené řízení obsahu* obsahuje vylepšené funkce správy, včetně správy metadat, sdílení služeb aplikací, lepší údržba, syndikace obsahu a politik.
- *Social computing* – možnost vytvoření plnohodnotné firemní sociální sítě (profily uživatelů, informační kanály).

Dále se SharePoint Server rozděluje na verze Standard a Enterprise, které se od sebe liší již pouze v detailech a v rozsahu nabízených služeb. Jak lze výše vidět, tyto verze jsou vhodné spíše pro rozsáhlé a komplexní řešení pro větší firmy a korporace.



Obrázek 4.1: Uspořádání vývojové platformy Windows SharePoint 2010. Nad Windows Serverem služby operačního prostředí (IIS,.NET Framework, SQL). Nad IIS běží ASP.NET, kde už běží jednotlivé WSS služby, které jsou přímo napojeny na .NET Framework [13].

4.3.1.3 SharePoint Designer

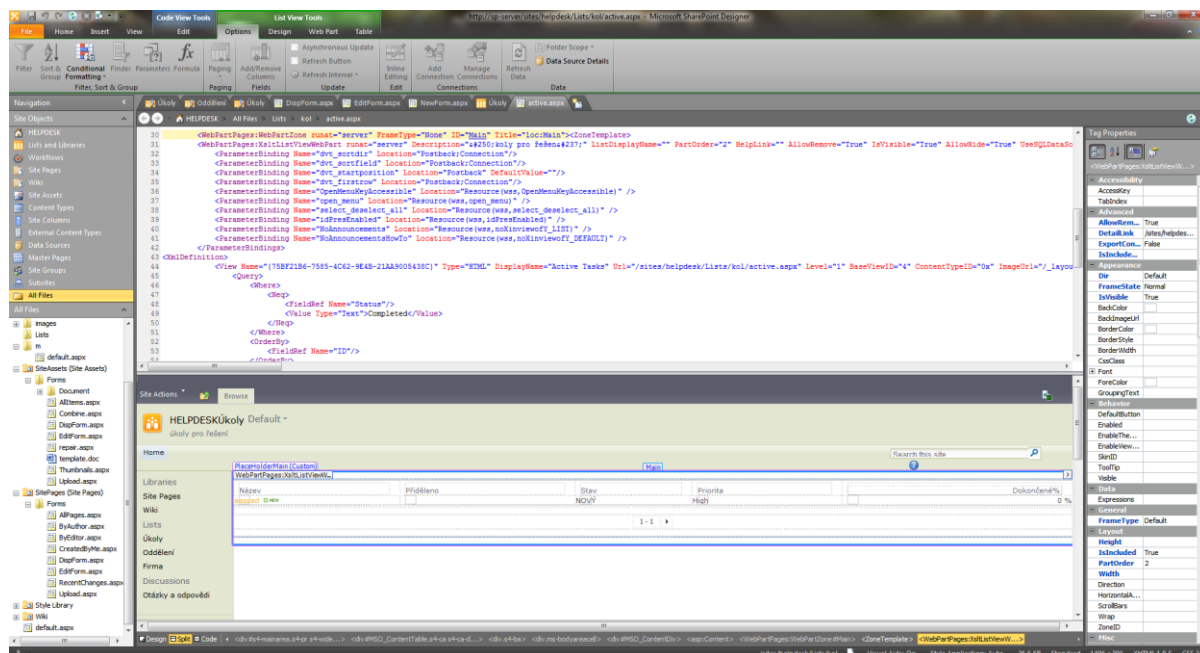
Je specializovaný HTML editor a web design editor pro vytváření a úpravy webu a stránek produktů SharePoint. Z vývojového hlediska byly jeho dřívější verze částečným nástupcem programu Microsoft Expression Web (dříve FrontPage, což byl software určený k editaci HTML kódu) zaměřený ke tvorbě v platformě SharePoint. Současná verze 2010 už nemůže pracovat samostatně bez ostatních produktů této platformy (Foundation nebo Server). První verze „2007“ tohoto produktu byla produkována jako komerční software, ale od roku 2009 je poskytován jako freeware.

Dále tento program umožňuje:

- Úpravu ASPX stránek, což jsou dynamicky generované webové stránky, které jsou vytvořeny ve frameworku pro webové aplikace ASP.NET (kap. 4.3.2), které se v SharePointu využívají.
- Práci s datovými zdroji (seznamy).
- Vytváření *pracovních postupů* prostřednictvím definovaných podmínek, pravidel a akcí.
- Generování sestav pomocí XSLT transformací (převod zdrojových dat ve formátu XML do libovolného požadovaného formátu jako HTML, případně jiných datových struktur).

Lze ho tedy využít jak k vytvoření/vložení nové stránky nebo kolekce webů, ale i k úpravě existujících stránek nebo formulářů. Jeho možnost pracovat s ASPX kódem, umožňuje mnohem širší využití, jako je vkládání webových částí a samotné programování. Možnost vytváření pracovních postupů nám také otevírá další možnosti využití, jde například o přidělování úkolů entitám a na základě výsledků a průběhu těchto úkolů řídit postup dále (změna, nový úkol).

Všechny tyto možnosti budou využitelné při vytváření Helpdesku.



Obrázek 4.2: Vzhled a vývojové prostředí aplikace SharePoint Designer 2010.

Na *obrázku 4.2* můžeme vidět rozmístění jednotlivých prvků a nástrojů programu. Můžeme na něm nalézt editor kódu a designu včetně podrobnějšího nastavení, jednotlivé objekty a soubory celé stránky.

4.3.1.4 SharePoint Workspace

Tento poslední produkt z rodiny SharePoint je také součástí Microsoft Office Professional Plus 2010 a dříve byl známý jako Microsoft Office Groove. Je to aplikace čistě určená ke tvorbě a synchronizaci dokumentů a spolupráci s více uživateli nebo týmem při jejich vytváření. Jeho základem je sdílený pracovní prostor, obsahující sadu souborů ke sdílení a některé nástroje ke skupinové spolupráci. Uživatelé, kteří mají potřebná oprávnění, navzájem sdílejí a upravují společný pracovní prostor, mohou mezi sebou komunikovat a synchronizovat potřebná data. Mají možnost offline i online synchronizace s určenými weby a dokumenty služby SharePoint, nebo případně spolupracovat s externími partnery, kteří nesdílí společnou infrastrukturu.

4.3.2 ASP.NET

Je to webový aplikační framework vyvinutý společností Microsoft, umožňující vytvářet dynamické webové stránky, webové aplikace a webové služby. Jde vlastně o rozšíření technologie ASP (Active Server Pages – nástroj k vytváření dynamických a interaktivních stránek) o programovatelnost pomocí technologie .NET, což nám umožňuje psát program ASP.NET využitím jakéhokoliv .NET programovacího jazyka (např. Java, C#, VB) [15]. To je možné díky CLR neboli Common Language Runtime, což je standart technologie .NET, definující prostředí pro spuštění nativního kódu. V CLR je kód vyjádřen pomocí CIL – Common Intermediate Language, do kterého se během kompilace překládají všechny kódy napsané v CLR kompatibilním jazyku.

Jinými slovy ASP.NET je skriptovací technologie běžící na straně serveru umožňující spouštět skripty vložené do webových stránek. Jedná se o program běžící přímo v IIS. Z hlediska souborů je využívána přípona **".aspx"** (případně asax, ascx, ashx, asmx, axd, soap při využití v různých nástrojích a verzích). Tyto soubory jsou zpracovávány na straně serveru a mohou obsahovat statické HTML, XML značky a skripty, které jsou spustitelné na serveru. Dynamický kód spouštěný na serveru může být do stránky vložen v rámci určeného dynamického bloku. Jeho spuštění může probíhat více způsoby. Pokud prohlížeč žádá HTML soubor, server vrátí tento soubor. Pokud prohlížeč žádá soubor typu ASP.NET, IIS předá žádost přímo ASP.NET engine na serveru, a tento vrátí prohlížeči obyčejný HTML soubor. ASP.NET engine čte soubor řádek po řádku a spouští jednotlivé skripty.

Jak můžeme vidět na *obr. 4.1*, služby technologie SharePoint běží přímo nad ASP.NET Frameworku, jeho použití se tedy přímo nabízí a není tedy žádný problém s tímto spojením. Stránky v SharePointu se ukládají také jako aspx soubory, lze je tedy bez problému přímo editovat v jiném nástroji.

4.3.3 Microsoft Visual Studio 2010

Jedná se o vývojové prostředí na platformách Microsoft Windows. Může být využito pro vývoj konzolových aplikací i aplikací s grafickým rozhraním spolu s Windows Forms, webovými stránkami, aplikacemi i službami. Konkrétně využiji hlavně komponentu a nástroj Web Designer, který je uzpůsoben pro vývoj a úpravu webových stránek a využívá se pro vývoj aplikací ASP.NET.

Protože program pro SharePoint lze vyvíjet i pomocí .NET, využiji i tohoto vývojového prostředí k editaci a tvorbě. A jelikož SharePoint Foundation běží v .NET frameworku a má veřejný objektový model, lze do něj doprogramovat téměř cokoli, jak samotné GUI, tak jednodušší komponenty, které bude nutné vytvořit pro práci. Můžeme tak přistupovat přímo k obsahu dat, vytvářet nebo upravovat vlastní komponenty, tvořit prvky uživatelského rozhraní, případně vkládat celé stránky nebo hotové aplikace. Zejména tak budu pracovat přímo s daty a zdroji z SharePointu.

Visual Studio podporuje všechny jazyky .NET, mohu tedy využít jakýkoliv. Pro spojení s kódem ASP.NET využívá model kódu na pozadí (značka - identifikátor v jednom .aspx souboru, programovací kód v souboru dalším, např. v .aspx.cs pokud je tento kód v jazyce C#). Výstupem Visual Studia při vývoji nad SharePointem je knihovna, která obsahuje třídy prostředí .NET, tyto mohou být zděděny a tedy i svázány s třídami objektového modelu SharePointu.

4.3.3.1 Webová část (Web part)

Prvek, který využiji ve vývoji. Jedná se o jednu z funkcí ASP.NET a jde o část HTML kódu, který mohu vkládat do *aspx* stránky jako její součást. Z hlediska vývoje je to zřejmě nejjednodušší způsob tvorby pro SharePoint. Pokud jde o hierarchii při vytváření webových částí v infrastruktuře ASP.NET, jedná se o zděděnou třídu ze třídy `System.Web.UI.WebControls.WebParts.WebPart`, obsahující důležité funkce a metody, která je součástí ASP.NET [21]. Tato třída funguje jako ovládací prvek (případně obsahující další podřazené prvky). Při vlastním vývoji, tak do webové části vloží své vytvořené prvky, anebo nahradím stávající metody vlastními. Ve výsledku se jedná o webovou část v HTML kódu, který lze vkládat do SharePoint webových stránek.

4.3.3.2 Event Receiver

Další důležitou komponentou, kterou lze pro SharePoint vytvořit a kterou v této práci využiji, je tzv. *Event Receiver*. Jedná se o zvolenou operaci, kterou můžeme spustit po nějaké události, která ve vytvořeném systému (databázi) nastane [20]. Tato operace je vyvolána po provedení nějaké další operace nad některou z položek v databázi SharePoint, případně nad ní samotnou. Vazby na Event Receivery jsou v podobě tříd SharePointu (`Microsoft.SharePoint.SPItemEventReceiver`). Každá z nich obsahuje metody, které určují druhy operací, které se nad danou složkou provádí. Lze je provádět již v průběhu operace nebo až po jejím dokončení (je tak možné například zastavovat průběh operací, případně měnit obsahy). Tyto metody můžeme nahradit vlastními kódy a tak s nimi dále

libovolně pracovat. Pomocí Event Receiverů budu schopen mimo jiné implementovat návrh expertního systému dle flowchart.

4.4 Možnosti vývoje aplikace

V mém případě využiji produkty SharePoint Foundation jako instalovanou komponentu nutnou k běhu na serveru a SharePoint Designer pro tvorbu programu a helpdesku. SharePoint Server nebude nutné využívat, protože obsahuje víceméně nadstandartní funkce hodící se spíše pro obecně obsáhlejší a komplexní řešení velkých společností, které jsou v jisté míře obsaženy i ve verzi Foundation. Navíc budu používat individuální procesy a funkce, které buď přidám jako podpůrné služby a komponenty, případně doplním/doprogramuji pomocí použitého programového rozhraní .NET.

4.4.1 Využití nástrojů

Při vývoji mohu využít vestavěných nástrojů, které jsou součástí SharePoint Foundation. Jejich možnosti jsou ale omezené a jde jen o základní funkce, poskytující jednoduché operace. Další možností je vkládání .NET modulů a vlastního obsahu do již vytvořené farmy webu, které vytvořím pomocí dodatečných vývojových nástrojů.

Protože k samotnému obsahu v SharePoint Foundation a jeho webovým aplikacím můžeme snadno přistupovat a pracovat s ním jako s běžným souborovým systémem, využiji k jeho editaci SharePoint Designer. Pomocí něj navrhnu vzhled celé aplikace a vytvořím základní i pokročilejší funkcionalitu. Tam kde již tento nástroj nebude postačovat, případně bude nutné vytvoření složitějších nebo individuálních funkcí, využiji možnost veřejného objektového modelu a mohu tak bez větších omezení vkládat vlastní .NET moduly (přístupy k objektům, vlastní funkce, vytváření ovládacích prvků, široká možnost propojení). V tomto ohledu použiji vývojové prostředí Visual Studio, kde mohu jednak vytvořit programy samotné a pak je vkládat, nebo přímo editovat SharePoint soubory. Je zde také možnost připojit přímo k webové aplikaci Foundation Serveru, podobně jako SharePoint Designer a vkládat kód přímo do ní.

4.4.2 Základní řešení vývoje

Jak je již zmíněno výše, technologie SharePoint Foundation umožňuje sama o sobě do jisté míry vytvářet jednoduché aplikace, které jsou postavené na operacích se seznamy a sestavami, pracovních postupech a zabezpečení.

Seznamy jsou zde zastoupeny jako množina sloupců, podobně jako tabulky v SQL. Každý z těchto sloupců je datovým typem (možnost výběru základních datových typů, případně vlastního) a jsou v něm uloženy odpovídající informace. Pokud vybereme datový typ sloupce jako odkaz na položku umístěnou v jiném seznamu, můžeme vytvářet vzájemné relace. Takto lze odkazovat i na

více položek a je tak možné vytvořit i relace typu M:N. Tyto seznamy mohou být několika typů. Jednak obecný seznam, který je základním typem a umožňuje uchovávání dat ve sloupcích. Všechny ostatní typy jsou rozšířením tohoto obecného seznamu a mohou to být například knihovny dokumentů, kalendáře nebo seznamy úkolů (seznamy položek, případně dokumentů určitých typů, jimž se přizpůsobuje samotné GUI). Rozšířené typy těchto seznamů se tak ve výsledku v klientských aplikacích chovají podle jejich vlastností. Na zdroje vytvořené z těchto seznamů je možné také vytvořit například jednoduché sestavy aplikováním XSLT transformací.

Dalšími operacemi jsou definované pracovní postupy. Jde například o doplňování a úprava informací, schvalování a další operace nad daty a úložišti. V základu jsou předem definovány jednoduché postupy, jako je schvalování obsahu a dokumentů, upozorňování uživatelů. Jsou ale použitelné pouze v základních případech, ve složitějších nebo dynamičtějších je nutné vytvořit jistou logiku, případně celý postup.

V SharePoint Foundation můžeme také na každý objekt podrobně nastavit přístupová práva. Lze tak každému webu, seznamu nebo položce přidělit nebo omezit přístup určitým uživatelům případně skupině (od možnosti samotného přístupu, čtení, úpravy, až k přidávání položek). Zabezpečení se může nastavovat na uživatelské účty a skupiny v Active Directory nebo na vytvořené skupiny v SharePointu (kde mohou např. i objekty Active Directory patřit do této skupiny). Navíc kromě samotného zabezpečení objektů můžeme provádět i audity přístupů, protože můžeme uchovávat informace o tom, kdo a kdy ke kterému dokumentu přistupoval a jaké operace byly prováděny.

Už jen na základě těchto základních funkcí jde vytvořit jednoduchý informační systém nebo webovou aplikaci s organizační strukturou, řízením práv a úložištěm dokumentů. Pro rozšíření o další pokročilejší funkce a operace, které bude potřeba pro požadovaný helpdesk vytvořit, ale využijí výše zmiňované a popsané nástroje. Podrobný postup vytváření celého systému je popsán v následujících kapitolách.

5 Implementace systému a testování

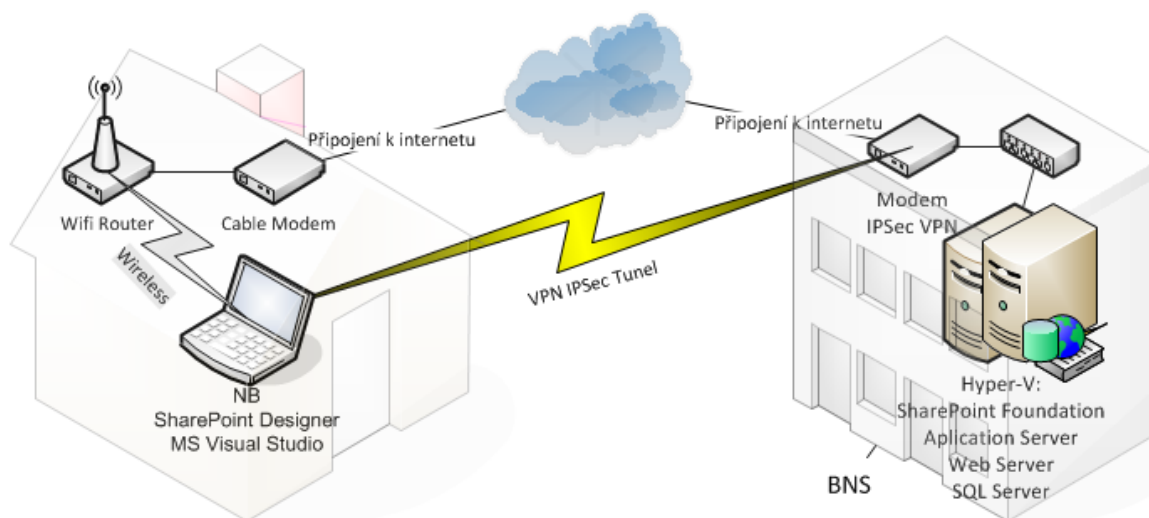
Počínaje touto kapitolou prakticky využijí veškeré znalosti uvedené v předchozích kapitolách, k vytvoření co nejlépe aplikovatelného helpdesku k lepší, rychlejší a optimalizovanější správě vzniklých problémů a požadavků určených pro firmu BNS. Celý problém je rozdělen do několika následujících podkapitol, a to podle postupu při vývoji. První fáze je vývoj a testování na vytvořeném vhodně nastaveném virtuálním serveru. Poté bude následovat přesun celé aplikace/serveru do samotné firmy a její podrobnější testování za provozu.

5.1 Instalace a nastavení implementačního prostředí

Pro vývoj aplikace v mém případě jsem využil prostředí jednoho ze serverů BNS, na kterém běží systém Windows Server 2008 R2. Zde jsem vytvořil virtuální server, který pracuje jako webový server. Potřebné vývojové programy jsem nainstaloval na lokální PC, ze kterého jsem helpdesk vzdáleně vyvíjel.

Pro předběžné testování jsem využil virtualizační software VMware Workstation, pomocí kterého jsem rozběhl zmíněný serverový systém na lokálním PC a odzkoušel si konfiguraci včetně všech nastavení a instalací. To vše tak, aby vše bylo funkční a nebyl žádný problém s následnou instalací a konfigurací přímo na serveru (nastavení a data lze jednoduše přenést). Na obrázku 4.3 je zobrazeno síťové zapojení při vývoji aplikace.

Co se týká doporučené konfigurace systémového prostředí, oficiální požadavky pro využití SharePoint Foundation na serveru, který tak bude plnit funkci webového a aplikačního serveru (společně s Active Directory a SQL serverem) jsou: 64bitový, 4 jádrový procesor, 4GB paměti RAM, prostor 80GB na systémovém disku [19].



Obrázek 4.3: Zapojení v síti při implementaci.

5.1.1 Server

Aplikaci budu vyvíjet pomocí jednoho serveru, na kterém budou zprovozněny potřebné aplikace, nástroje, funkce a jednoho klienta s vývojovými nástroji. Proto byl, po testování na lokálním počítači (s nástrojem VMware), na jeden ze serverů BNS se systémem Windows Server 2008 R2 nainstalován virtuální server pomocí virtualizační technologie Microsoft Hyper-V Server R2. Ta je již integrovanou součástí zmíněného OS jako nástroj, který stačí nakonfigurovat podle potřeb.

Podrobnější konfigurace serveru je popsána níže. Dalším krokem je instalace služby SharePoint Foundation, při níž je nutné nastavit prostředky, které budu využívat. Po instalaci je potřeba vytvořit novou webovou aplikaci, ke které se lze připojit vzdáleně pomocí vývojových nástrojů.

Pokud budeme uvažovat běh serveru pro testovací účely a pro začátek menší zátěž z uživatelské strany, není problém umístit všechny potřebné služby (doménový řadič, web server, SQL server) na jeden aplikační server. Po úspěšném nasazení serveru do reálného provozu a vytížení v několika firmách bude již vhodné rozdělit tyto služby mezi jednotlivé fyzické servery (jak také momentálně běží).

5.1.1.1 Nastavení, konfigurace a služby serveru

Po samotné instalaci serveru musí být nakonfigurován do určitých rolí a funkcí. Pro běh webové aplikace SharePoint proto pomocí konfigurace v *Server Manager* přidám serveru roli *Web Server (IIS)*, *Application Server* a také roli doménového řadiče (*Domain Controller*) organizovaného pomocí *Active Directory*. Dále musí být nainstalována služba *SQL server 2008*. Kompletní výčet spuštěných rolí a funkcí je v příloze na obrázku 7.1 a 7.2.

5.1.1.2 Active Directory

SharePoint Foundation fungují v prostředí Active Directory a pomocí jejich vlastností můžeme řídit práva a přístupy jednotlivých uživatelů k aplikacím bez nutnosti zakládání uživatelských účtů přímo v SharePoint. Vytvořený server s názvem SP-SERVER jsem přidal do domény `bns.local` (v případě testování ve VMware si vytvořil doménu `sp.local`).

Pro běh serverů a aplikací je doporučeno používat vlastní servisní účty, z důvodů zabezpečení i správné funkčnosti jednotlivých služeb. Proto si pro moje účely vytvořím následující nové účty zvlášť pro SharePoint a SQL server.

- doména\SQL_Admin (Servisní účet serveru SQL)
- doména\SP_Admin (Účet pro připojení do farmy)
- doména\SP_Service (Servisní účet služeb SharePoint a IIS)
- doména\SP_Search (Účet pro přístup vyhledávací služby SharePoint)

5.1.1.3 Databáze

Pro vytvoření a správu databáze jsem využil Microsoft SQL Server 2008 R2. K běhu SharePoint aplikace postačuje instalace základních nástrojů a klientských komponent spolu komponenty databázových služeb. Databázový server je nastaven tak, aby se uživatelé mohli přihlašovat pomocí vytvořených účtů v Active Directory. Při instalaci DB server byl zvolen název instance *SQL_SP*, také vytvořen uživatelský servisní účet *SQL_Admin*. (po migraci do firmy bude připojen firemní databázový server)

5.1.1.4 Windows PowerShell

Jedná se o rozšířený shell (interpret příkazů - příkazový řádek) se skriptovacím jazykem a je vestavěnou součástí použitého Windows Serveru. Je založen na platformě .NET Framework. Poskytuje proto pomocí specializovaných tříd velké množství funkcí pro správu [25]. Je také implementován pro podporu v různých aplikacích např. jako *SharePoint 2010 Management Shell*. Využijí ho hlavně pro spouštění a úpravu vlastností potřebných služeb. Využijí ho zejména ke tvorbě skriptů sloužících pro testování spojení se zařízeními a doplňování výsledků do seznamu.

5.1.1.5 SharePoint Foundation

Součástí samotné instalace jsou i potřebné komponenty a aktualizace, které aplikace vyžaduje k běhu a které je potřeba před samotnou instalací na serveru mít. Následně je nutné zvolit, zda se bude SharePoint Foundation instalovat samostatně (stand-alone), nebo jako webové uživatelské rozhraní (které zvolím) a umístění ukládaných datových souborů. Po samotné instalaci je prostředí připraveno pro konfiguraci a nastavení a vytvoření nové farmy. Konfiguraci programu a vlastností vytvářené aplikace mohou provádět pomocí *SharePoint Central Administration* (viz obr. 7.4).

5.1.1.6 Vytvoření a spuštění nové webové aplikace

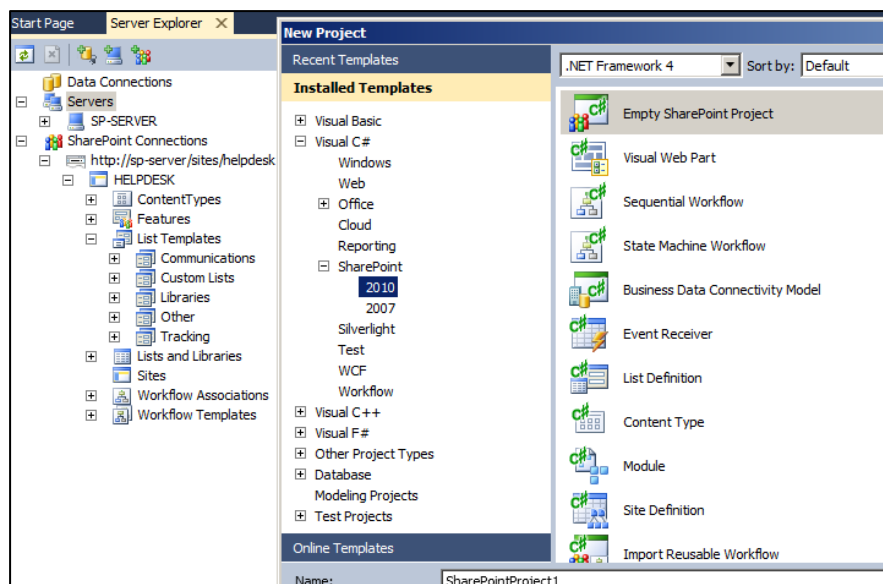
Po instalaci SharePoint Foundation, je spuštěn konfigurační průvodce. Máme možnost se buď připojit k existující webové farmě, nebo vytvořit novou. V mém případě zvolím vytvoření nové farmy, čímž vytvořím i novou webovou aplikaci poskytující základní služby a konfiguračním rozhraním, se kterým mohu dále pracovat. Vytvořená webová aplikace slouží ke konfiguraci celé vytvořené farmy. Následně jsem pomocí konfiguračního průvodce umístil konfigurační databázi farmy na vytvořený databázový server (SP-SERVER/ SQL_SP) a pojmenoval ji defaultně *SharePoint_Config*. Také je nutné nastavit uživatelský účet pro přístup k této databázi. Pro připojování na webové rozhraní jsem pro připojení na administrativní rozhraní zvolil port 81 protokolu http (pro provoz samotné aplikace je využit default port 80). Jako autentizační protokol jsem nastavil výchozí NTLM (NT Lan Manager) společnosti Microsoft standardně používaný v OS Windows.

Po úspěšně dokončené konfiguraci jsem následně přímo při spuštění programu vytvořil webovou aplikaci pojmenovanou *sp-server.sp.local* a umístil na ni prázdnou kolekci webů helpdesku, která má adresu <http://sp-server/sites/helpdesk>. Dalším krokem je spuštění potřebných služeb pro běh aplikace. Ty lze nastavit v administrativním rozhraní *SharePoint Central Administration – Services on server* na serveru. Všechny potřebné služby jsou spuštěny již ve výchozím nastavení (všechny spuštěné služby na obr. 7.3 v příloze). Navíc spustím službu *SharePoint Foundation Search*. Ta zaznamenává informace o obsahu vytvořených webů, ukládá je do databáze, kterou při spuštění služby nakonfiguruji, a následně je využívá a poskytuje při hledání na webu. Jako název databáze zvolím výchozí *SharePoint_Search*, a v Active Directory vytvořím uživatelský účet *SP_Search* pro přístup k datům této databáze.

5.1.2 Nástroje pro vývoj

Všechny nástroje mám k dispozici na lokálním počítači, z něhož se budu přihlašovat vzdáleně přes VPN do firmy a následně na aplikační server. *Sharepoint Designer* je k dispozici ke stažení zdarma. Po jednoduché instalaci je možné se připojit/otevřít (pomocí zadání umístění) web stránku, kterou jsem si vytvořil a nastavil na serveru pomocí SharePoint Foundation. Po otevření je možné ji vytvářet a editovat. Vzhled vývojového prostředí Designeru můžeme vidět na *obr. 4.2*. Samotný program obsahuje i editor stránek, kde lze jak vkládat nové tak upravovat stávající (přednastavené). Můžeme v něm samozřejmě i přímo pracovat se zdrojovým kódem *aspx* a tak do stránek vkládat vlastní vytvořené webové části nebo je zde programovat. V Designeru vytvořím kompletní tělo celého heldesku, které budu následně podle potřeby upravovat a přidávat nutné prvky.

Dalším programem, kterého využiji jako klienta je *Visual Studio 2010*, kde je postup podobný jako u SharePoint Designeru. Tento vývojový program je vhodné nainstalovat přímo na server (pro přímé spojení s aplikací SharePoint je nutná instalace na stejném počítači). V základním nastavení jsem zvolil možnost práce jako *Web Developer*. Po spuštění aplikace se pak stačí připojit k SharePoint Foundation Serveru a následně upravovat jednotlivé komponenty. Nebo přímo vytvořit nové webové části (*obr. 4.4*), které budu vkládat do vytvořeného webu (*viz 4.3.3*). Další možností je samozřejmě samostatná úprava a doplnění zdrojového kódu souboru *aspx* vytvořeného v SharePointu.



Obrázek 4.4: Vytváření ve Visual Studiu, připojení na SharePoint server.

5.1.3 Klienti

Zastupují zákazníky, kteří budou helpdesk využívat. Připojovat se budou obdobně jako na *obr. 4.3*. Případně bude stránka zpřístupněna na některém webu firmy bns (např. www.bns.cz/helpdesk). Mohu počítat s tím, že na všech klientských počítačích běží Windows XP Professional SP3, případně

Windows 7 Professional, je nainstalován kancelářský software MS Office 2007 a v systému je obsažen webový prohlížeč Internet Explorer 7 /8 a .NET Framework 3.5 klient. Uživatelé se budou moci z jejich sítě k helpdesku přihlašovat zadáním webové adresy do prohlížeče, která je přesměruje na webovou aplikaci na serveru, kde jsou následně přihlášení pod svým uživatelským účtem.

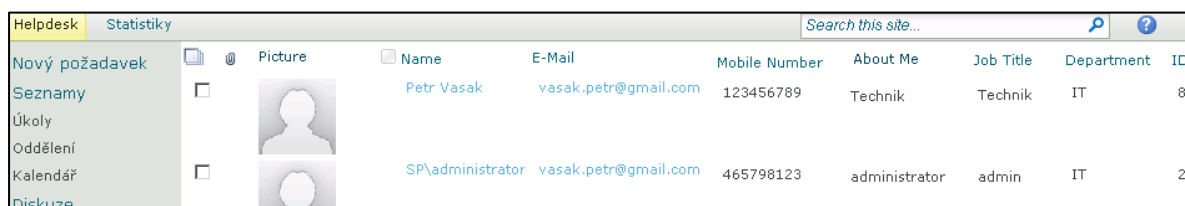
5.2 Návrh systému



Po instalaci a nastavení nutných programů a komponent bude samotný helpdesk vytvářen pomocí navrženého flow chart diagramu, který bude základem. Dále je nutné si určit uživatelské role, vytvořit doménový model s popisem využitých datových struktur a definovat si funkčnost a logiku postupů včetně báze znalostí, podle které se bude mimo jiné řešení odvíjet.

Úkolem je vytvořit systém helpdesk, dle definovaných doporučení, pro efektivnější a rychlejší řešení zadávaných a vzniklých problémů, které se vyskytují ve spravovaných firmách. Zároveň se zpětnou odezvou, umožňující zpracování a vyřešení části nebo celého problému automaticky, pomocí předem učených postupů a znalostní databáze. Každý požadavek bude mít svoje zařazení a prioritu, to vše podle informací zadaných při jeho vytvoření (příp. doplnění). Je tedy nutné si definovat potřebné specifikace, podle kterých budu systém vytvářet. Jde hlavně o funkce a postupy aplikace, logiku vytvořených akcí, popis uživatelských rolí a datových struktur.

5.2.1 Uživatelské role

Pro využívání systém a přidělení práv uživatelům je nutné si definovat uživatelské role. Každý uživatel tak bude mít jistou roli a s ní spojená práva. To znamená, jaké akce může uživatel provádět případně k jakým datům a umístěním bude mít přístup. Uživatelské role definuji pomocí uživatelských skupin SharePoint Foundation. Uživatelé do nich budou integrováni z Active Directory, v nichž jsou již na firemních serverech všichni uloženi. Odpadá zde tedy nutnost znovu pokaždé vkládat všechny uživatele a všechny informace o uživatelích jsou tak automaticky přeneseny. Na vytvořeném webu pak budou zobrazeni jako na *obr. 4.8*. Uživatelské role jsou následující:



	Picture	Name	E-Mail	Mobile Number	About Me	Job Title	Department	ID
<input type="checkbox"/>		Petr Vasak	vasak.petr@gmail.com	123456789	Technik	Technik	IT	8
<input type="checkbox"/>		SP\administrator	vasak.petr@gmail.com	465798123	administrator	admin	IT	2

Obrázek 4.8: Zobrazení uživatelů v helpdesku.

5.2.1.1 Administrátor

Uživatel, který má veškerá možná práva, která lze v aplikaci provádět. Správce a kontrola nad celým systémem. Může vkládat, mazat nebo měnit veškeré seznamy, knihovny a dokumenty. Dále má možnost vytvářet, přidávat, měnit a odstraňovat všechny uživatele a skupiny.

5.2.1.2 Řešitelé

Všichni uživatelé firmy BNS. Mohou vkládat nové úkoly (pro případ telefonického požadavku), dále mají možnost řešení, změn a mazání úkolů patřících do jejich oddělení případně jim samotným (tedy obchodníci, technici, reklamace, programátoři). Ostatní úkoly mohou prohlížet. Také mají možnost přispívat do knihoven s dokumenty, do „wiki page“, „discussion board“ (viz kapitola. 5.3.1.8) a prohlížet statistiky helpdesku. Programátoři mohou navíc upravovat vzhled a jednotlivé prvky.

5.2.1.3 Vedoucí firmy

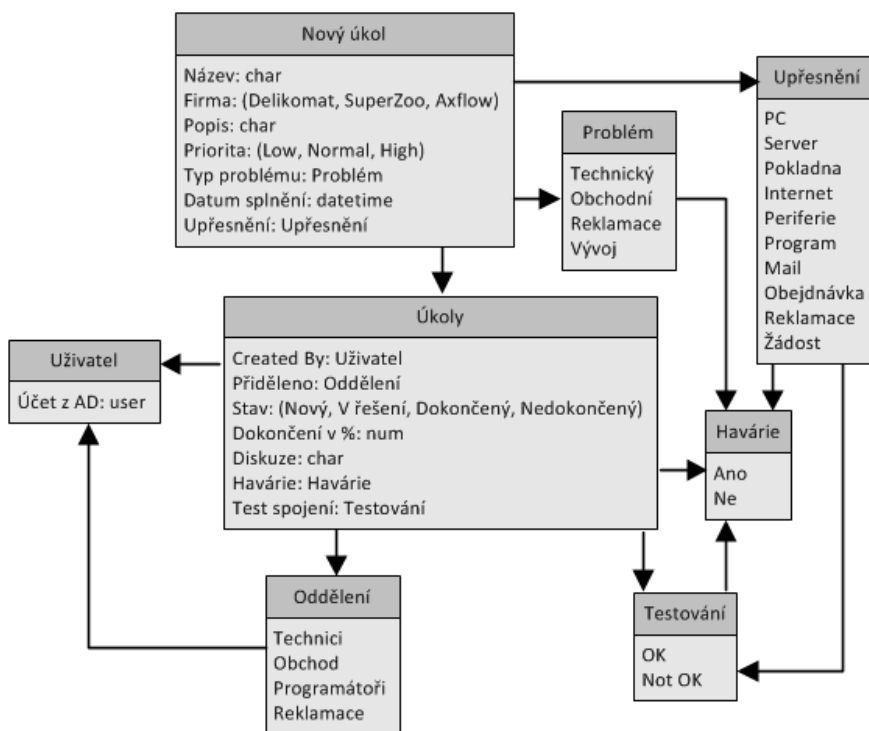
Představuje uživatele, který je ředitelem nebo vedoucím ve spravované firmě. Má tedy kontrolu nad požadavky svých zaměstnanců. Má práva úkoly rušit, měnit, a zadávat nové požadavky v rámci vlastní firmy. Má možnost čtení knihoven a dokumentů a zadávat otázky v discussion board.

5.2.1.4 Zadavatel

Zaměstnanci jednotlivých spravovaných firem. Mohou zadávat nové požadavky a problémy, případně svoje zadání dodatečně upravovat (odpovědi v diskuzi požadavku). Mohou vidět a prohlížet všechny požadavky v rámci dané svojí firmy. Dále mohou prohlížet knihovny a dokumenty a zadávat dotazy.

5.2.2 Model databáze

Na obrázku 4.5 můžeme vidět diagram doménového modelu aplikace včetně popis datových struktur, které jsou v aplikaci využity. Jsou v něm současně zobrazeny vztahy při akcích v seznamu. Následně jsou jednotlivé položky popsány podrobněji.



Obrázek 4.5: Doménový model databáze helpdesku.

5.2.2.1 Úkol

Seznam všech problému k řešení, které byly zadány uživateli systému. Při zadávání každého úkolu musí uživatel vyplnit povinné položky (výběr z několika možností + popis problému), na základě kterých je úkol vyhodnocen a přidělen. Patří mezi ně název, priorita, firma, typ problému, jeho upřesnění a požadované datum dokončení. Po vložení úkolu do seznamu je rozšířen o zadavatele, datum vložení, přiděleného řešitele, stav (ve kterém se řešení nachází), diskuzi problému a konečně na základě zadání je vyhodnoceno, zda se jedná o havárii a jestli je funkční spojení se serverem.

5.2.2.2 Oddělení

Položka definující oddělení do jakého úkol spadá a podle toho i přidělená skupina řešitelů úkolů, kteří mají přidělen úkol v systému a jsou o něm informováni. Oddělení, tedy řešitelé jsou zvoleni automaticky po vložení a podle typu úkolu, nebo dodatečně změněn samotným řešitelem podle uvážení.

5.2.2.3 Problém a upřesnění

Problém definuje, jakého odvětví se požadavek týká. Podle toho je přidělen řešitel. Upřesnění navíc udává z několika možností, čeho se požadované řešení problému týká. Na základě těchto údajů jsou provedena další vyhodnocení.

5.2.2.4 Priorita

Určuje důležitost přidělenou úkolu. Lze tak rozeznat, co je nutno řešit prioritně a co může počkat na později. Priorita je rozdělena do více úrovní, aby tak byla možnost řešení přehlednější.

5.2.2.5 Havárie a Testování

Určují, zda se jedná o havárii, tedy problém, který nějakým způsobem brání provozu, nebo běžný úkol, který je možné řešit s větším časovým odstupem. Testování ve vybraných případech uvádí, zda se podařilo spojit se službou, která je předmětem řešeného problému (server, internet, mail).

5.2.2.6 Uživatel

Definuje uživatele – zadavatele a řešitele, vyskytující se ve vytvořených skupinách aplikace, s přidělenými právy. Jedná se o uživatele z Active Directory.

5.2.3 Funkce helpdesku

Následuje popis jednotlivých operací a probíhajících událostí, které je v aplikaci možné provádět. Také popis prováděných akcí, které v systému budou během jeho pracovních cyklů probíhat.

5.2.3.1 Možnosti uživatelů

Všichni uživatelé využívající aplikaci jsou obsaženi v databázi aplikace společně se svými uživatelskými rolemi, a tedy i oprávněními k určitým operacím a přístupem do částí helpdesku. To vše podle předchozího rozdělení uživatelských rolí a po samotném přihlášení každého uživatele do aplikace.

5.2.3.2 Zadávání úkolů

Pokud chce uživatel vložit do systému nový úkol, spustí se nové okno s výběrem požadovaných informací a průvodcem při zadávání problému viz *obr. 4.7*. Uživatel si zde vybere z nabídek, které definují právě jeho problém (typ služby, úkolu, priorita, popis, vložení přílohy, atd.) a co nejpřesněji ho tak specifikuje. Během tohoto výběru, v závislosti na zadaných parametrech, zde budou vypisovány informace (odkazy na stránku s nápovědou, případně diskuzi), které mohou pomoci uživateli vyřešit problém samostatně, ještě než ho zadá do systému.

Po samotném zadání úkolu je tento vyhodnocen dle zadaných parametrů a vložen do seznamu úkolů. Následně je tak provedeno přidělení řešiteli, zaslání informačního emailu a dle logiky diagnostika funkčnosti sítě a zařízení s doplněním dodatečných údajů. V systému je pak tento úkol zobrazen řešiteli s kompletními informacemi. Všechny podrobnosti uživatel o úkolu zjistí po otevření náhledu v podobě seznamu nejpodstatnějších informací jednotlivých úkolů.

5.2.3.3 Typy a priorita úkolů a služeb

Všechny úkoly v systému lze rozdělit podle priorit, které jsou přiděleny v době zadávání úkolů (případně po úpravě řešitelem). Pro tuto funkci je vytvořen seznam priorit úkolů. Pokud je již při zadávání zvolena jistá kombinace parametrů (například havárie) je úkolu automaticky přidělena priorita nevyšší pro okamžité řešení.

5.2.3.4 Řešení úkolů

Po vložení úkolu do seznamu je připraven pro řešení přiděleným řešitelem a je označen jako *nový*. Tento stav může řešitel v životním cyklu daného úkolu měnit na *řešený*, *odložený* nebo *hotový*. Při zvolení poslední možnosti se úkol uzavře a je zaslán informační email zadavateli o ukončení. Dále má řešitel možnost doplňovat popis úkolu, přidávat doplňující informace nebo dotazovat zadavatele, případně předávat problém jinému oddělení řešitelů. Co se týká dodatečného dotazování zadavatele, je uskutečněno zadáním dotazu do příslušného boxu, což je doprovázeno informováním zadavatele emailem. Jakmile zadavatel odpoví, je řešitel stejným způsobem informován.

5.2.3.5 Databáze znalostí

Může být součástí řešení našeho helpdesku. Jsou v ní uloženy nejčastější problémy uživatelů daného systému a jejich řešení. Je využívána zejména tam, kde se opakovaně vyskytují stejné nebo podobné

problémy. Při využití znalostní databáze tak stačí zadavatele úkolu o této skutečnosti informovat. Následně je tak při zadávání nového úkolu (při daném výběru) uživatel systémem (případně řešitelem) vhodně upozorňován a teprve v případě, že problém pomocí těchto nápověd sám nevyřeší, zadá ho do systému.

5.2.3.6 Shrnutí a výstupy

Informace o stavu seznamů, akcí a řešených úkolech. V aplikaci jsou zastoupeny oddělenou sekci. Zde je obecný přehled o počtu a řešení úkolů, případně různých vhodných statistik pro zjištění stavu seznamů, uživatelů a helpdesku. Administrátorovi, případně řediteli, je tak umožněn dohled nad aplikací a jejím stavem a vytížením.

5.2.4 Postupy a logika řešení

Následuje popis samotných postupů a dějů v programu při různých operacích s databází a jejich rozhodovací logiky při řešení. Při tvoření helpdesku jich budu využívat hned několik, jsou to následující.

5.2.4.1 Vytvoření databáze

Pro vytvoření obecných seznamů databáze použiji funkce, které jsou standardně součástí SharePoint Foundation. Pomocí nich vložím všechny potřebné seznamy, které jsem si definoval v této kapitole výše. Na vytvořenou databázi podobným způsobem také aplikuji přístupová práva. Pokud bude u některých seznamů nutné automaticky nastavovat přístupová práva na jejich jednotlivé položky nebo spustit nějakou akci v závislosti na změnách v databázi, využiji v tomto případě již zmiňované Event Receivery. Ty si vytvořím podle potřebné funkčnosti, například pro nastavení práv při změně určité položky. Ke vkládání, zadávání a úpravy všech potřebných seznamů využiji standardní formulář aplikace, který si eventuálně upravím podle potřeb.

5.2.4.2 Logika při zadávání úkolů

Při vytváření úkolů využiji rozhodovací logiky. Ta bude na základě zadaných údajů doplňovat další podle vytvořených pravidel (báze znalostí). Bude tak schopna přiřazovat řešitele úkolů, doplňovat další nevyplněné položky, nabízet zadavateli určité možnosti z databáze znalostí a v neposlední řadě testovat dostupnost a připojení částí sítí a zařízení, kterých se úkol týká. Pro všechny tyto akce bude nutné také vytvořit vlastní Event Receivery, které budou spuštěny po vložení záznamu do databáze. Pro vytvoření a editaci každého úkolu bude využit jeden přizpůsobený formulář (*obr. 4.7*), uzpůsoben pro co nejpřehlednější a nejpřesnější zadání úkolu.

5.2.4.3 Postupy

V celé aplikaci bude vytvořen pracovní postup, který bude spuštěn při vložení nového úkolu do databáze a jeho vyhodnocení (viz výše). Každý řešitel bude mít po přihlášení pod svým účtem seznam přidělených úkolů (také bude informován o vložení nového). Po ukončení řešení některého z těchto úkolů se řešení uzavře a vytvořený pracovní postup je tím také ukončen. K vytváření pracovních postupů a jejich spouštění po vytvoření úkolu využiji funkce SharePoint Designeru.

5.2.4.4 Statistiky a reporty

K tomu, abych mohl generovat přehledové statistiky a reporty, lze využít generování dokumentů pomocí XSLT transformace, které jsou součástí SharePoint Designeru, společně s obsaženými nebo individuálně vytvořenými webovými částmi. Pomocí nich mohu zobrazovat souhrny informací o seznamech, uživatelích a procesech v systému.

5.3 Vytváření systému

Pro vyvíjený systém je v jedné farmě vytvořena nová kolekce webů ve webové aplikaci SharePoint. Celá tato aplikace tak bude izolována od ostatních a zároveň jednodušeji přenositelná s možnostmi následných úprav, nastavení a záloh. Veškeré nastavení serveru a potřebných programů je popsáno v kapitole 5.1. Nyní implementuji popsané řešení z kapitoly 5.2.

5.3.1 Implementace

Samotnou implementaci jsem začal nastavením a spuštěním potřebných služeb viz výše. Pro vývoj a implementaci jsem si na testovacím virtuálním serveru vytvořil kolekci webů a zvolil umístění helpdesku do `http://sp-server/sites/helpdesk`, kde byla umístěna nová prázdná stránka kolekce webů. Všechno tedy vyvíjím od začátku, bez šablon nebo přednastavených komponent. Následuje popis úprav a změn v rámci vytvořené kolekce webů určenou pro vývoj aplikace.

5.3.1.1 Struktura a vzhled

Prvním krokem bude přizpůsobení vzhledu systému a jeho struktury. Jak je již zmíněno, vše bude přidáváno do prázdné stránky. Na obrázku 4.6 je výsledná podoba titulní stránky helpdesku ve webovém prohlížeči po přihlášení. Celá stránka je vytvořena z prvků a částí dostupných v SharePoint Designeru a možností SharePoint Foundation. Po otevření a přihlášení k systému bude mít každý uživatel přehled o svých požadavcích a ostatního obsahu (otázky, dokumenty). Tento titulní obsah se bude automaticky při změnách obnovovat, takže nebude nutno obnovovat stránku nebo zacházet do jednotlivých oddílů pro zjištění nějaké změny. Dále na *obr. 5.4* můžeme vidět zobrazení helpdesku na mobilním zařízení (testováno na telefonu s OS Android). Veškeré nastavení vzhledu stránky i jejich jednotlivých prvků (včetně přidávání vytvořených řešení a webových částí) mohu upravovat přímo

v SharePoint Designeru nebo v natavení samotné stránky v prohlížeči (obr. 7.7). A to buď grafickou úpravou designu a rozmístění komponent a webových částí, nebo podrobněji doplňováním a změnou zdrojového kódu. Těmito způsoby jsem přizpůsobil výsledný vzhled a doplnil potřebné části.

5.3.1.2 Uživatelé

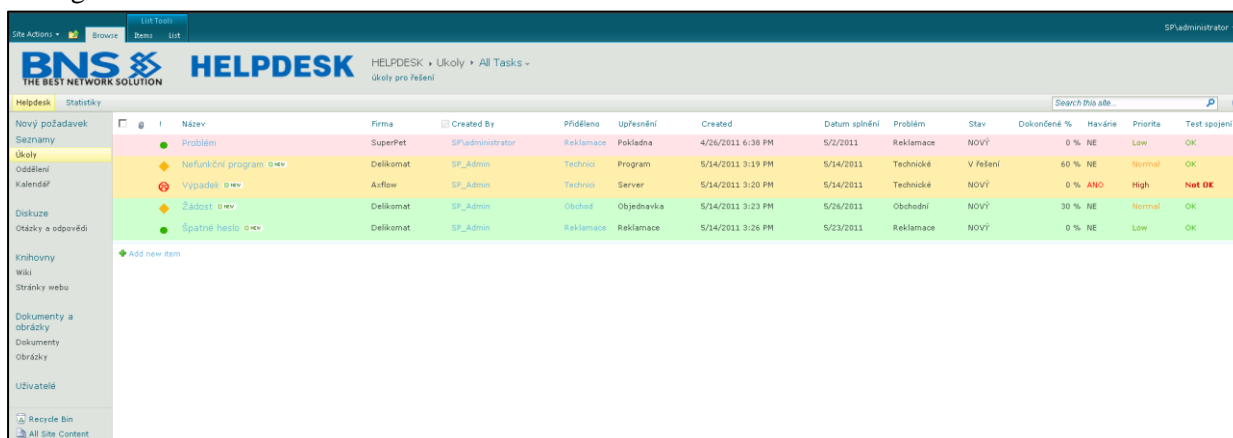
V Active Directory jsem si vytvořil zkušební data v podobě uživatelských skupin zastupující uživatelské role, které jsem si definoval. Do nich jsem vložil několik uživatelských účtů. Při testování a běhu v reálném provozu toto nebude potřeba, pouze se přidělí reálné účty a skupiny do uživatelských skupin v SharePointu. Ty jsem si v něm vytvořil čtyři, tedy *Administrátoři* (použitý při vývoji aplikace), *Řešitelé* (obsahuje skupiny *technici*, *obchod*, *reklamace*, *programátoři*), *Vedoucí*, *Zadavatelé*. K řízení přístupu ve webové aplikaci je nutné přidělit těmto skupinám jistá oprávnění v rámci kolekce webů. Pro *administrátory* jsem nastavil plné řízení, pro *řešitele* „contribute“ (přispívání) a pro *vedoucí* a *zadavatele* pouze čtení. Toto je pouze obecné nastavení, všude tam, kde bude potřeba oprávnění změnit (jednotlivé seznamy a položky), je možné ho přesně nastavit.

5.3.1.3 Seznamy

Všechny potřebné seznamy, které jsem si definoval v modelu databáze pro práci, jsem vytvořil přímo v SharePoint Designeru. Všem seznamům jsem pak nastavil oprávnění podle potřebných přístupů uživatelských skupin (například vedoucí má možnost editovat požadavky svých zaměstnanců).

5.3.1.4 Formuláře

Nejdůležitější formulář představuje okno vkládání úkolů. Všechny formuláře pro úpravu a vytváření jednotlivých položek seznamů jsou vytvořeny automaticky. V mém případě doplním formulář pro vytvoření úkolu o informace určené zadavateli, kde bude upozorněn na databázi znalostí, obsahující řešení častých problémů a jejich možných řešení bez kontaktu podpory. Následně může zadavatel přejít přímo k formulaci problému. Toto řešení je provedeno úpravou struktury formuláře vkládání nových položek (*NewForm.aspx*) daného seznamu úkolů v editoru stránek kolekce webů v SharePoint Designeru.



The screenshot shows the BNS HELPDESK application interface. The header includes the BNS logo and the text 'HELPDESK THE BEST NETWORK SOLUTION'. Below the header, there is a navigation menu on the left with options like 'Nový požadavek', 'Seznamy', 'Úkoly', 'Oddělení', 'Kalendář', 'Diskuze', 'Otázky a odpovědi', 'Knihovny', 'Wiki', 'Stránky webu', 'Dokumenty a obrázky', 'Uživatelé', 'Recycle Bin', and 'All Site Content'. The main content area displays a table of tasks with columns: 'Název', 'Firma', 'Created By', 'Přiděleno', 'Upřesnění', 'Created', 'Datum splnění', 'Problém', 'Stav', 'Dokončené %', 'Havárie', 'Priorita', and 'Test spojení'. The table contains several rows of task data, including 'Problém', 'Nefunkční program', 'Výpadek', 'Žádost', and 'Špatné heslo'.

Název	Firma	Created By	Přiděleno	Upřesnění	Created	Datum splnění	Problém	Stav	Dokončené %	Havárie	Priorita	Test spojení
Problém	SuperPet	SP_Administrator	Reklamace	Pokladna	4/26/2011 6:38 PM	5/2/2011	Reklamace	NOVÝ	0 %	NE	Low	OK
Nefunkční program	Delkomat	SP_Admin	Technica	Program	5/14/2011 3:19 PM	5/14/2011	Technická	V řešení	60 %	NE	Normal	OK
Výpadek	Avflow	SP_Admin	Technica	Server	5/14/2011 3:20 PM	5/14/2011	Technická	NOVÝ	0 %	ANO	High	Not OK
Žádost	Delkomat	SP_Admin	Obchod	Objednávka	5/14/2011 3:23 PM	5/26/2011	Obchodní	NOVÝ	30 %	NE	Normal	OK
Špatné heslo	Delkomat	SP_Admin	Reklamace	Reklamace	5/14/2011 3:26 PM	5/23/2011	Reklamace	NOVÝ	0 %	NE	Low	OK

Obrázek 4.6: Rozhraní aplikace HelpDesk.

Protože v tomto programu jsou vcelku omezené nebo žádné možnosti nastavení zobrazení jednotlivých prvků ve formulářích (např. v zadávání nového úkolu), spravoval jsem potřebná zobrazení pomocí skriptů v PowerShellu. Ukázka možnosti nezobrazení jednoho z polí seznamu (pouze pro čtení) je v příloze 2.

5.3.1.5 Logika a aplikace expertního systému

Jde o veškeré funkce, které se budou provádět automaticky. Ty jsou jednak implementovány podle navrženého diagramu aplikace a jednak jsou to potřebné menší operace přímo na webu. Jedná se o jednodušší prvky jako samostatný výběr řešitele, zasílání mailů, určení havárie nebo nastavení práv uživatelů pro seznamy a operace, které jsou řešeny přímo v SharePointu pomocí pracovních postupů a formulí. Dále to jsou operace, jako přidělení barvy dle akutnosti, pokročilé nastavení oprávnění skupinám nebo kontrola vzdáleného připojení, které jsou podrobněji popsány v následující kapitole vývoje a implementace. Navržené techniky expertních

systému realizují pomocí pracovních postupů (workflows) a úpravou aspx kódu, dále pak vytvořením webových částí a Event Receiverů.

The screenshot shows a 'Ukoly - New Item' form. At the top, there's a toolbar with 'Edit', 'Format Text', and 'Insert' tabs, and buttons for 'Save', 'Cancel', 'Paste', 'Copy', and 'Attach File'. Below this is a 'Commit' button and a 'Clipboard' section. The form fields include: 'Název' (Name), 'Firma' (Company) with a dropdown, 'Popis' (Description) with a text area, 'Priorita' (Priority) with a dropdown, 'Problém' (Problem) with a dropdown, 'Datum splnění' (Due Date) with a date picker, 'Upřesnění' (Clarification) with a list of categories (PC, Server, Pokladna, Internet, Periferie, Program, Mail, Objednavka, Reklamacie, Žádost), 'Stav' (Status) with a dropdown, 'Dokončené %' (Completed %) with a percentage input, 'Diskuze' (Discussion) with a text area, and 'Přiděleno' (Assigned to) with a user selection dropdown. A 'Save' button is at the bottom right.

Obrázek 4.7: Okno pro zadání nového úkolu.

5.3.1.6 Pracovní postupy

Pracovní postupy neboli „Workflows“ jsou jednou ze součástí SharePointu. Pomocí nich můžeme například přiřadit akce seznamu (při vložení nového řádku, manuálně, při změně), která se bude vykonávat. Mohou mít i několik kroků. Hlavní pracovní postup bude vytvořen v závislosti na seznamu úkolů. V pracovních postupech se budou provádět dodatečné akce, čili informování zadavatelů a řešitelů mailem (s podrobnými informacemi o stavu jejich úkolu), na jejich adresu uvedenou v AD, při vytvoření, změně požadavku (doplnění dotazu, změnu řešitele) a jeho ukončení.

Aby bylo možné odesílat maily, je samozřejmě nutné nastavit v *Central Administration* SMTP server odchozí pošty (mail server BNS) a mailovou adresu, ze které budou zprávy odesílány. Je také možné přidat a nakonfigurovat službu SMTP do funkcí IIS serveru (naš server pak bude fungovat jako virtuální SMTP server). Na *obrázku 7.5* v příloze 1 je definovaná automatický akce zaslání mailu zadavateli a řešiteli při zadání nového úkolu včetně jeho formy. Obdobně budu řešit i přiřazení úkolu odpovídajícímu řešiteli (*viz obrázek 7.6*), nebo změnu stavu úkolu po modifikaci. Po jakékoliv změně požadavku budou mít uživatelé možnost manuálně odeslat upozornění (které bude obsahovat aktuální diskuzi – dodatečné dotazy a odpovědi) relevantním uživatelům, a to tlačítkem, které je k dispozici při označení potřebného řádku seznamu (*obr. 5.2*). Při dokončení úkolu se zpráva odešle automaticky (a úkol se nebude nadále zobrazovat v seznamu).

Pracovní postupy budu vytvářet také buď pomocí Event Receiverů, kde tak mohu vytvářet akce bez omezení (testování spojení, spouštění skriptů), nebo sekvenčními postupy (Sequential Workflow), kdy jsou v obou případech dostupné předdefinované šablony ve Visual Studiu 2010. Také je možné importovat vytvořené postupy z aplikace Microsoft Visio 2010.

5.3.1.7 Formule

Přímo v SharePoint Designeru jdou při vyplňování vložit prvky seznamu (sloupce), které mohou být nějakým způsobem závislé nebo vypočítané v závislosti na jiných vložených sloupcích. Tyto hodnoty lze zadávat formulemi. Pomocí nich můžu vytvořit jednoduché i složitější podmínovací, matematické, textové i datové funkce, pomocí kterých je vytvořena výsledná hodnota [22]. Mohu tak jednodušeji a bez použití event receiverů vytvořit funkční části stránek (v rámci jednoho seznamu). Využiji toho například při stanovení, zda se jedná o havárii (při zadání vysoké priority nebo problému se serverem). Uvedu příklad formule pro určení, zda problém spadá do havárií.

```
=IF(OR(AND([Priorita]="High", [Upřesnění]="Server", [Problém]="Technické"),
AND([Priorita]="High", [Upřesnění]="Mail", [Problém]="Technické"),
AND([Priorita]="High", [Upřesnění]="Internet", [Problém]="Technické"),
AND([Priorita]="High", [Upřesnění]="Pokladna", [Problém]="Technické"),
[Test spojení]= "Not OK"), "ANO", "NE")
```

Tedy pokud je ve sloupci s názvem „*Priorita*” hodnota „*High*”, zároveň ve sloupci „*Upřesnění*” problém týkající se serveru, mailu, internetu nebo pokladny, a zároveň je „*Problém*” technického rázu, nebo pokud neuspěl test spojení se serverem, vrát sloupci „*Havárie*” hodnotu „*ANO*”, v opačném případě vrát „*NE*”. Podobným způsobem mohu získat i další potřebné hodnoty.

5.3.1.8 Databáze znalostí a dokumenty

Zde využijí vestavěných „wiki page“ knihoven, kde budou moci administrátoři a technici zapisovat časté problémy (FAQ) a uživatelé na ně budou odkázáni při zadávání svých problémů. Případně v podobě generického seznam s názvem „discussion board“, kde se mohou koncoví uživatelé pokládat dotazy a řešitelé na ně mohou odpovídat. Na prvky v této databázi lze odkazovat i pomocí pracovních postupů. Přímou na stránce bude také odkaz na úložiště dokumentů a obrázku, pro případnou pomoc a návody i sdílení dokumentů je možné využít i tyto knihovny. Jejich vzhled je v příloze 1 na obr. 7.10.

5.3.1.9 Výstupy a reporty

Pro výstupy a statistiky helpdesku jsem si vytvořil samostatnou stránku dostupnou z hlavní lišty s informativním přehledem údajů o využívání aplikace (*statistics.aspx*). Následně jsem ve Visual Studiu vytvořil (pomocí SharePoint tříd) a vložil na tuto stránku webovou část, zobrazující statistiky a informace o seznamech a uživateli. Ukázka kódu je uvedena v příloze 2 (*VisualWebPart1UserControl.ascx.cs*). Na obrázku 5.5 je část aktuálních statistik přístupných na webu.

Informace o seznamu úkolů

Vlastník	Info	Popis
Autor	SP\administrator	Uživatel, který vytvořil seznam
Vytvořeno	4/19/2011 1:58:36 PM	Datum a čas vytvoření
změněno	5/16/2011 6:44:51 PM	Datum a čas poslední změny
položek v seznamu	16	Celkový počet úkolů v seznamu

Statistiky a upozornění na změny uživatele

Akce	Info	Popis
Uživatelů celkem	0	Celkový počet uživatelů provádějící změny
Počet změn	0	Počet všech změn prováděných nad seznamy
Přidávání	0	Počet přidávaných položek
Změny	0	Počet změn nad seznamy
Mazání	0	Počet smazaných seznamů nebo položek

Seznam změn dokumentů

Položka	Typ změny	Datum a čas
Nefunkční program	Add	5/14/2011 1:19:07 PM
Nefunkční program	Update	5/14/2011 1:19:41 PM
Nefunkční program	Update	5/14/2011 1:19:58 PM
Nefunkční program	Update	5/14/2011 1:20:08 PM
Výpadek	Add	5/14/2011 1:20:27 PM
Výpadek	Update	5/14/2011 1:20:33 PM

Obr. 5.5: Ukázka stránky se statistikami.

Také jsem zde vložil dodatečně nainstalovanou webovou část nazvanou *Report Viewer*, která využívá služeb *Microsoft SQL Reporting Services* (součást instalovaného SQL Serveru) a soubory *Report Definition Language* (RDL). Tyto reporty je možné vytvořit například ve Visual Studiu. K běhu *Report Vieweru* bylo potřeba povolit službu *Session State* (příkazem `Enable-SPSessionStateService -Defaultprovision` v PowerShellu) a v IIS pro moji webovou aplikaci přidat modul této služby s názvem `System.Web.SessionState.SessionStateModule`. K zobrazení těchto reportů, stačí zvolit jejich umístění (*rdl* souboru). Zároveň je možné využít generování (výstupních) dokumentů pomocí XSLT transformací v SharePoint Designeru.

5.4 Řešení vytvořených prvků

Doplnění těla aplikace vytvořeného v SharePoint Designeru a SharePoint Foundation o dodatečně vložené prvky zdrojového kódu a webových částí k řešení pokročilejších problémů. Tyto funkce budou tvořit v SharePoint Designeru a ve Visual Studiu úpravou *aspx* souborů, a zároveň přitom využijí vkládání nových webových částí a event receivery [26].

5.4.1 Postup vývoje a aplikování jednotlivých komponent

5.4.1.1 Vytvoření a vložení webové části

Novou webovou část můžeme do editovaných stránek vložit buď přímo v SharePoint Designeru (v případě editace aspx souborů). Další možností je vytvoření a následné doplnění o požadované funkce ve Visual Studiu v kódu libovolného .NET jazyka, v mém případě v C#. Poté můžeme tuto již hotovou webovou část vložit do stávajících stránek v SharePoint. Podrobněji popíši postup vytvoření konkrétní použité webové části.

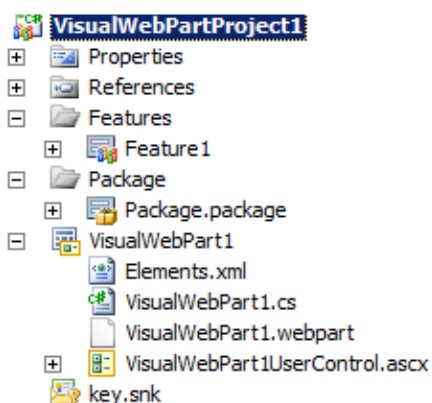
Nejdříve vytvoříme ve Visual Studiu 2010 nový projekt (pro spojení s SharePointem musí být nainstalován přímo na serveru). Nabídka možností je na *obr. 4.4*. Tedy zvolíme jazyk a v něm jednu ze šablon SharePoint 2010. To bude v mém případě *Visual Web Part* (případně dále *Event Receiver*). Po výběru je nutné uvést lokální SharePoint stránku, tedy `http://sp-server/sites/helpdesk`. Následně je vytvořen nový projekt a všechny potřebné soubory, připraveny k editaci (viz *obr. 5.1*). Ukázka z ladění projektu je v *příloze 1* na *obr. 7.9*. Nejdůležitější soubory pro vývoj jsou:

VisualWebPart1UserControl.ascx – obsahující `VisualWebPart1UserControl.ascx.cs`, jsou to ovládací prvky, poskytnuté SharePoint instancí. Tyto budu editovat. Lze přidávat nástroje „drag and drop“ formou.

VisualWebPart1.cs – třída webové části, obsahuje zdrojový kód pro danou webovou část, využívá se k řízení a personalizaci nastavení.

Elements.xml – konfigurační soubor popisující strukturu a obsah, potřebný při tvorbě balíčku webové části.

Obsah jednotlivých zdrojových souborů, konkrétně webová část pro vypsání informací o seznamu úkolů, je uveden ve zdrojových kódech v *příloze 2*.



Dalším krokem je přeložení tohoto projektu a testování. Pokud ho následně i spustím, otevře se přímo webová stránka, na kterou byl projekt navázán a je zde možnost webovou část přímo vložit a otestovat. Jakmile běh projektu ukončím, je tato možnost deaktivována a řešení je odstraněno z lokální SharePoint instance. Pokud dále chceme tuto webovou část nasadit, použijeme soubor `.wsp` obsahující sestavenou webovou část (uloženou po překladu ve složce *bin* projektu) [24].

Obrázek 5.1: Složení webové části.

K instalaci řešení do SharePointu se použijí následující příkazy v *SharePoint 2010 Management Shell*:

```
Add-SPSolution C:\WebPart.wsp
```

- Instalace webové části (pokud je např. umístěna na C:\)

```
Install-SPSolution -identity WebPart.wsp -WebApplication
```

```
http://sharepointweb -GACDeployment
```

- Pokud je webová část již nainstalována, takto ji lze updatovat novou verzí

Následně pak stačí naplnit galerii webových částí touto vytvořenou (vybrat a zvolit). Poté je tato webová část připravena k přidání na SharePoint web (obsažena mezi ostatními webovými částmi).

5.4.1.2 Vytvoření a aplikování Event Receiverů

Tuto možnost využijí pro spouštění složitějších operací, po nějaké události, která vznikne nad seznamem (případně jiným prvkem systému). Stejně jako v předchozím případě vytvoření webové části, při vytváření nového projektu ve Visual Studiu, vybereme šablonu *Event Receiver*. Následně zadáme SharePoint web, na který ho chceme aplikovat. Dále je nutné zvolit, jakého typu tento event receiver bude (list events, list item events, web events,...) a jaký bude zdroj a typ události na který bude reagovat. Pro moje účely (viz dále) jsem zvolil, že půjde o událost nad položkou seznamu úkolů, při jejím přidání. Po vytvoření projektu stačí editovat nově vzniklý soubor (s názvem *EventReceiver.cs*) a doplnit potřebné operace, které se budou provádět. Po spuštění projektu je vytvořený event receiver nasazen v galerii řešení na odpovídajícím webu, takto ho lze otestovat. Pro ostré nasazení je nutné v *SharePoint 2010 Management Shell* spustit následující příkazy (podobně jako pro webovou část):

```
Add-SPSolution "C:\EventReceiver.wsp"
```

```
Install-SPSolution EventReceiver.wsp -GACDeployment
```

Následně stačí přidat do galerie řešení (podobně jako webovou část – soubor typu *wsp*) v nastavení konkrétního webu a aktivovat toto řešení. V *Central Administration* je také nutné spustit službu *Microsoft SharePoint Foundation Sandboxed Code Service*, která je nutná k ladění a spouštění event receiverů [27].

5.4.1.3 Pokročilé nastavení oprávnění uživatelů

Všechny oprávnění jednotlivých uživatelů se nastaví podle popisu v předchozí kapitole s popisem uživatelů. Obecná práva na jednotlivé seznamy a knihovny jdou nastavit přímo na vytvořené stránce pro jednotlivé skupiny. Pokud ale chceme nastavit přístup k jednotlivým položkám v seznámech (např. aby zadavatelé viděli pouze svoje úkoly a řešitelé úkoly svojí skupiny), je nutné vytvořit

odpovídající event receiver. A to tak, že nové přidělení práv se bude spouštět po zachycení události přidání nebo změny položky v seznamu úkolů. Při ní se všechny nastavení přístupových práv vymažou a pro danou položku seznamu je zadavateli a řešiteli přiděleno právo změny, ostatní (kromě administrátora) ji mohou číst. Podobným způsobem mohou přidělovat unikátní práva uživatelům helpdesku pro libovolnou část, při nějaké události. Obsah vytvořeného event receiveru je v *příloze 2*.

5.4.1.4 Rozdělení dle barev

Důležitým prvkem zejména pro celkovou přehlednost je barevné zobrazení seznamu s jednotlivými úkoly. Toto rozdělení jsem aplikoval přímo na daný seznam a jeho jednotlivé prvky dopsáním odpovídajícího kódu (skripty v souboru *AllItems.aspx* daného seznamu). Lze tak provést doplněním podmíněných kritérií v programu (tzv. Conditional formatting, *obr. 7.8 v příloze 1*).

Označeny budou jednak barevně celé řádky, a to podle data splnění (před vypršení, blíže vypršení, po vypršení). Samotný kód pro řádek, pak vypadá takto:

```
<xsl:attribute name="style">
    <xsl:if test="ddwrt:DateTimeTick(ddwrt:GenDisplayName(string($thisNode/@StopDate)))
    &lt;ddwrt:DateTimeTick(ddwrt:GenDisplayName(string($Today)))"ddwrt:cf_explicit="1">
        background-color: #FDD7D7;</xsl:if>
    <xsl:if test="$Collapse">display:none;</xsl:if>
</xsl:attribute>
```

- Pro případ, že datum splnění je menší, než aktuální datum (po vypršení).

Dále odliším určité položky barevně a řádky úkolu barevným znakem podle toho, jakou mají prioritu:

```
<xsl:if test="$thisNode/@prior = 'n'" ddwrt:cf_explicit="1">font-family: Wingdings; font-
weight: bold; color: #36B000; font-size: large;</xsl:if>
```

- Pokud prvek v seznamu nabyde určité hodnoty, změní se jeho styl.

Vzhled seznamu s úkoly má pak následující podobu:

	Název	Firma	Created By	Přiděleno	Upřesnění	Created	Datum splnění	Problém	Stav	Dokončené %	Havárie	Priorita	Test spojení
●	Problém	SuperPet	SPAdministrator	Reklamacie	Pokladna	4/26/2011 6:38 PM	5/2/2011	Reklamacie	NOVÝ	0 %	NE	Low	OK
◆	Nefunkční program <small>NEW</small>	Delikomat	SP_Admin	Technici	Program	5/14/2011 3:19 PM	5/14/2011	Technické	V řešení	60 %	NE	Normal	OK
⊗	Výpadek <small>NEW</small>	Axflow	SP_Admin	Technici	Server	5/14/2011 3:20 PM	5/14/2011	Technické	NOVÝ	0 %	ANO	High	Not OK
◆	Žádost <small>NEW</small>	Delikomat	SP_Admin	Obchod	Objednávka	5/14/2011 3:23 PM	5/26/2011	Obchodní	NOVÝ	30 %	NE	Normal	OK
●	Špatné heslo <small>NEW</small>	Delikomat	SP_Admin	Reklamacie	Reklamacie	5/14/2011 3:26 PM	5/23/2011	Reklamacie	NOVÝ	0 %	NE	Low	OK

Obrázek 5.2: Zobrazení seznamu s úkoly.

5.4.1.5 Testování připojení

Testování spojení bude probíhat v závislosti na tom, jaký zadá uživatel problém při vložení úkolu. Pokud se bude jednat o technický problém a zároveň se bude týkat mailu nebo serveru, provede se automatické otestování pomocí příslušného skriptu. A to buď na dostupnost samotného serveru v síti, nebo mailových služeb (mail server, služby příchozí a odchozí pošty). Testovat se bude pomocí

parametrů podle toho, jaké firmy a služby se bude problém týkat. Například pokud půjde o firmu Delikomat, a bude se jednat o technický problém mailu případně internetu, bude proveden odpovídající pokus o spojení na následující (porty dle individuální konfigurace serverů a služeb):

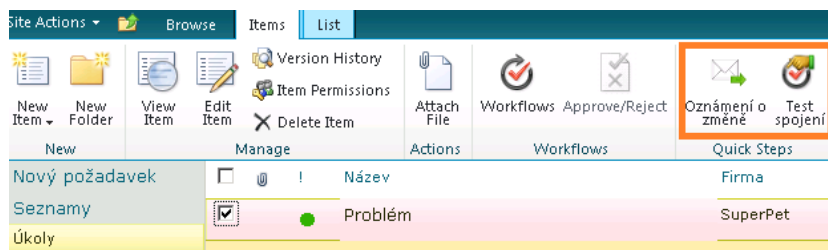
proxy.delikomat.cz 3389	- dostupnost RDP služby, tedy i spojení se serverem
proxy.delikomat.cz 443	- dostupnost OWA, tedy i poštovního serveru
proxy.delikomat.cz 25	- dostupnost a funkčnost odchozí pošty (SMTP)

Obdobně bude provedeno i u ostatních firem (případně služeb). Na základě úspěchu/neúspěchu těchto příkazů, se přímo v požadavku po jeho zadání zobrazí, zda je požadovaná služba dostupná nebo není.

Přidáním každého úkolu, který vyžaduje test spojení, je spuštěn testovací skript, který následně v závislosti na úspěšnosti doplní do úkolu informaci, zda je spojení v pořádku. To je dosaženo pomocí kombinace spuštěného pracovního postupu po přidání úkolu, spolu se skriptem vytvořeným v PowerShellu. Celý skript je v *příloze číslo 3*. Pro samotné spojení je využit příkaz:

System.Net.Sockets.TcpClient(\$remoteHost, \$port)

Otestování se provádí automaticky po přidání úkolu, ale bude jej také možné spustit manuálně z nabídky akcí pro každý úkol, pro případ že by bylo nutné otestovat spojení znovu po nějakém zásahu (stejně jako v případě informování o změně mailem, *obr. 5.3*).



Obr. 5.3: Akce úkolů
(Quick Steps).

5.4.1.6 Akce po přidání nového úkolu

Po přidání nového požadavku do seznamu úkolů se automaticky spustí vytvořené pracovní postupy. Jedná se o přidělení úkolu odpovídajícímu řešiteli, označení zda se jedná o havárii a odeslání informativní zprávy na mail uživatelů, kterých se úkol týká. Dále se seznam požadavků barevně označí dle akutnosti a data splnění. Každé nově vytvořené nebo změněné položce jsou vytvořeny práva pro přístup jednotlivých uživatelů. Pokud to okolnosti vyžadují (dle požadavku), je zkontrolována dostupnost a funkčnost služeb vzdáleného serveru.

5.5 Testování

Dalším krokem je otestování všech funkcí a služeb, které helpdesk vykonává a dále odzkoušení nasazení aplikace do reálného provozu ve firmě.

5.5.1 Migrace na firemní server BNS

Při přenosu celé aplikace na firemní server mohu použít dva možné postupy. Jednak je možné přenést celý „image“ serveru z programu WMware na server ve firmě, kde ho aplikuji pomocí virtualizační technologie Hyper-V. A to pomocí souboru image ve formátu VHD, který získám konvertováním z VMDK vytvořeného ve WMware. Zde následně přidám server do firemní domény BNS a nastavím firemní SQL server v konfiguračním nástroji SharePoint Foundation (pro možnost získání uživatelů z firmy včetně kontaktních informací, které následně rozdělím do vytvořených skupin v SP dle příslušnosti).

Dalším způsobem migrace je pouze přesun webové farmy, včetně všech částí potřebných pro její funkčnost. Tento postup je značně náročnější, protože je nutné nainstalovat a nastavit celý server znovu tak, jak jsem postupoval při vývoji a poté farmu importovat. Tato migrace je v tuto chvíli vhodná zejména pro instalaci nového fyzického aplikačního a webového serveru. Tento postup je popsán v *příloze 3: Manuály*.

V mém případě využiji první postup a to přenos image serveru. Stačí tedy přidat server do domény firmy a přidělit reálné uživatele z Active Directories přidané domény do odpovídajících skupin, které jsou již vytvořeny v SharePointu. Změnou domény se zároveň změní i všechna nastavení, která jsou potřebná pro běh programu. Pokud z nějakého důvodu nemůžeme změnit doménu serveru, je možné také zpřístupnění databáze uživatelů vyskytující se v jiné doméně. A ty je pak standardně možné přidávat do SharePoint skupin. Toto zpřístupnění je možné pomocí tzv. *Domain Trust*, což je způsob jak umožnit uživatelům z důvěryhodné domény přistupovat ke službám důvěřující doméně [29]. Je možné toho dosáhnout pomocí nástroje *Active Directory Domains and Trusts* (součást Windows Serveru 2008) tak, že se nejdříve nastaví DNS a delegování na cílovou doménu a na obou stranách se vytvoří dva stejné doménové administrátorské účty, pomocí kterých vzájemně zpřístupní Active Directory. Konkrétně pak jde o tranzitivní vazbu.

Posledním krokem je pak zveřejnění helpdesku na firemních stránkách (dostupných lokálním i vzdáleným uživatelům). To je možné publikováním adresy kolekce webu s helpdeskem na webovém serveru, kde běží stránky BNS.

5.5.2 Migrace aplikace na nový server

Pokud bychom uvažovaly přenos vytvořeného helpdesku včetně veškerého jeho obsahu a funkcí na nově nainstalovaný server, postup bude značně složitější, protože bude nutné od začátku provést veškerá nastavení tohoto severu a instalaci potřebných prvků a aplikací. Tedy nejdříve na Windows Server 2008 R2 zprovoznit funkce webového a aplikačního serveru, nainstalovat programy SQL Server 2008 a SharePoint Foundation, včetně podrobného nastavení. Podrobně, krok za krokem, je tento postup vysvětlen v *příloze 3*. Kde je nejdříve popsán postup instalace a nastavení serveru a následně migrace vytvořené aplikace webové farmy s helpdeskem na tento server.

Pokud bychom chtěli přenést samotnou kolekci webů, je možné udělat její zálohu v *SharePoint 2010 Management Shell* případně v *Central Administration*.

Příkaz pro vytvoření zálohy je:

```
Backup-SPSite -Identity http://server/site -Path "c:\backup.bak"
```

Pro obnovení pak stačí vytvořit prázdnou kolekci a umístit na ni zálohu následovně:

```
Restore-SPSite -Identity http://myserver/site -Path "c:\backup.bak"
```

Po obnově by měla být aplikace již funkční v případě, že se jedná o server využívající stejný Active Directory. V opačném případě je nutné nastavit administrátora webu a naplnit jednotlivé skupiny uživateli.

5.5.3 Postup testování a dosažené výsledky

Po prvním kroku samotného zprovoznění aplikace a dostupnosti ve firmě, jsem postupoval tak, že jsem existujícími uživatelskými účty naplnil skupiny v aplikaci. Zároveň jsem naplnil seznamy testovacími daty – úkoly. Vyzkoušel jsem jak korektnost a samotný průběh zadávání včetně automatizovaných řešení a akcí uskutečněných techniky. Dále možnosti jednotlivých uživatelů vzhledem k jejich přiděleným právům. Po odladění několika objevených vad, jako byly možnosti uživatelů s různými právy a nekorektní akce po přidání úkolů, byl systém připraven k testování v reálném provozu. Otestoval jsem tedy funkčnost systému v rámci možností firmy i s ostatními uživateli, abych tak pokryl co největší rozsah jeho možností.

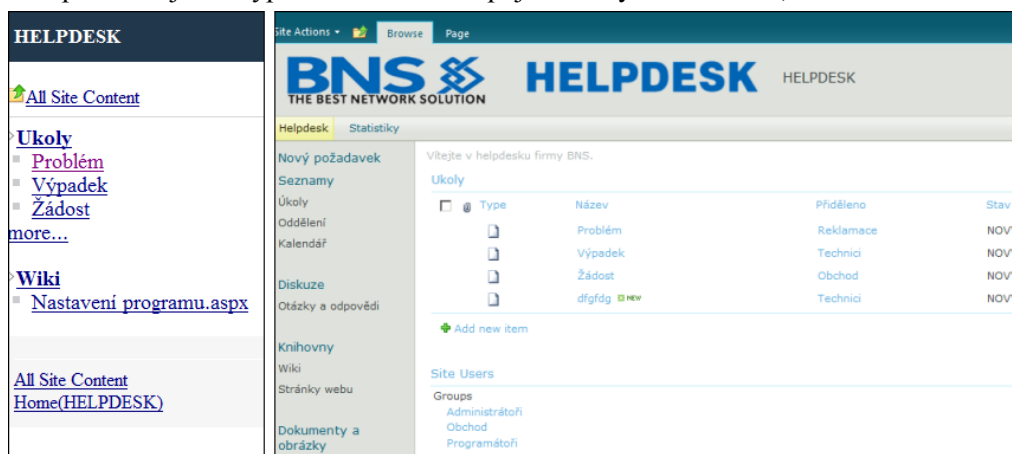
Celý systém nakonec pracoval v pořádku, a to podle navrženého flow-chart diagramu systému. Některé jeho kroky byly mírně pozměněny, kvůli realizačním možnostem, které nebyly při návrhu úplně známy a v průběhu řešení se rozvíjely v závislosti na aplikacích. K uplatnění helpdesku tak zbývá pouze naplnění veškerými pracovními daty (uživateli ze všech spravovaných firem s případnými dokumenty a knihovnami s návody) a zaběhnutí ve firmách společně s případnými dalšími rozšířeními (viz dále).

5.6 Další rozšíření a customizace produktu

Dalším krokem v rozvoji současného stavu aplikace bude přizpůsobení zbytku firem, které BNS spravuje. Půjde hlavně o doplnění nastavení dle topologií sítí ostatních firem a naplnění uživatelských skupin o odpovídající zaměstnance. Případně rozšířit některé funkce a položky seznamů, které nejsou prozatím obsaženy a v dalších společnostech by byly z hlediska korektního fungování požadovány.

I dále existuje nespočet možných rozšíření, která by obohatila funkčnost aplikace a umožnily tak další rozvoj. Zde vyberu ty nejzajímavější z hlediska dalšího přizpůsobení tohoto produktu společně s migrací do více firem.

Například je možné jednoduchým nastavením v SharePoint Foundation uzpůsobit helpdesk možnosti zobrazení na mobilních telefonech a zařízeních. Nastavení zahrnuje povolení mobilního zobrazení pro vybrané stránky, upravení zobrazovaných prvků, nebo určení adresy pro mobilní web (standardně nastaveno <http://sp-server/sites/helpdesk/?Mobile=1>). Na *obrázku 5.4* můžeme vidět dva vzhledy zobrazení na těchto zařízeních. Tato možnost je v dnešní době velice vítaná a praktická (například pokud dojde k výpadku sítě nebo odpojení firmy od internetu).



Obrázek 5.4: Vzhled helpdesku na mobilních zařízeních. Vlevo mobilní zobrazení, vpravo klasické.

5.6.1 Propojení s MS Exchange, Outlookem a Sociální sítě

Téměř všechny spravované firmy využívají poštovní řešení MS Exchange a tedy i MS Outlook nejen jako poštovního klienta, ale také jako firemní kalendář a pro sdílení veřejných složek. Proto se přímo nabízí využít možné propojení produktu SharePoint s Outlookem. Integrací Outlooku, kterou je možné provést přímo na vytvořených stránkách, lze získat synchronizaci s knihovnami dokumentů, seznamem kontaktů, úkolů a kalendářem. Celá aplikace se tak stane ucelenější a vznikne lepší přehled zejména nad úkoly. Dalším krokem v rozšiřování tímto směrem je návrh a implementace, dnes velmi oblíbených a diskutovaných, sociálních sítí. V tomto případě se však bude jednat pouze o firemní síť. Při vytvoření nových projektů nabízí SharePoint několik šablon aplikace, které jsou určeny přesně pro tyto účely. Jde zejména o rozšířené (tzv. sociální) profily jednotlivých uživatelů, které kromě kontaktních informací obsahují i různé stavy, diskuze a aktivity. Navíc je zde možnost vytváření

pracovních skupin, komunit, blogů a možnosti kolaborace na projektech (to vše s individuálním nastavením oprávnění). Můžeme tak v rámci jedné aplikace vytvořit firemní a pracovní obdobu známých sociálních sítí.

5.6.2 Širší komerční využití

V rámci naší republiky, v oblasti helpdesk systémů není žádný významnější, komerčně nabízený produkt nebo řešení s pokročilými funkcemi (zpětná odezva nebo samostatná diagnóza). Navíc většina menších firem orientující se na informační technologie nevyužívá žádná podobná řešení. Z tohoto důvodu by bylo zajímavé nabízet vývoj aplikace pro správu na míru i dalším firmám, a to stejným způsobem, jako byl vyvinut tento helpdesk (spojení SharePoint Foundation/Designer společně s .NET). Samozřejmě je třeba počítat s tím, že cílová firma musí využívat podobné řešení jako BNS, tedy MS Windows Server, SQL Server. Potom bude využívání produktu v rámci této firmy pro vlastní účely zdarma, a zákazník tak zaplatí pouze za vývoj, a ne navíc za licenci produktů/vývojových nástrojů. Helpdesk se potom přizpůsobí přesným požadavkům a zdrojům zákazníka, s možnostmi automatického vyhodnocování nebo testování, propojení s Outlookem, nebo vytváření reportů a auditů z celé spravované sítě.

6 Závěr

Cílem této práce bylo vytvořit pomocí vhodných nástrojů funkční helpdesk pro vzdálenou IT správu firem. Celý systém byl přitom stavěn „na míru“ firmě BNS, s tím, že hlavní důraz byl kladen na automatickou diagnostiku zadaných požadavků a vylepšení stávajícího řešení.

První teoretickou část práce jsem navázal na první dva body zadání převzaté ze semestrálního projektu. Formuloval a shrnul jsem známé a zadané požadavky a navrhl funkční model aplikace s využitím vhodných metod expertních systémů. Následná obsáhlejší část - praktická a realizační, zahrnovala výběr, instalace a nastavení vhodných nástrojů v závislosti na požadavcích, jejich vzájemné propojení a celkovou implementaci programu. Během práce jsem se musel vypořádávat s řadou problémů. Mezi ty hlavní patřilo vzájemné nastavení serveru a vývojových nástrojů, nekompatibilita jednotlivých nástrojů, kdy bylo nutné využít některého z alternativních řešení a mnohé chyby při vývoji, které si vyžádaly odlišný přístup k řešení.

V závěru práce bylo úspěšně provedeno testování a nasazení systému do provozu cílové firmy. V konečné diskuzi výsledů, se jevila jako reálná, možnost nabídnutí a aplikování vytvořeného produktu i v dalších podobných firmách, včetně rozšíření aplikace o řadu užitečných funkcí. Úspěšně jsem tak splnil a vypracoval všechny body zadání práce včetně vytyčených cílů. Navíc jsem se naučil a získal řadu cenných znalostí v jedné z oblastí vývoje informačních systémů (SharePoint, .NET) a serverových řešení (Windows Server 2008).

Literatura

- [1] Čejka, R.: *Návrh, správa, bezpečnost: Provoz a správa aplikací*. FIT VUT Brno, 2010.
- [2] Price, E., Hudson, K.: *Active Directory Domain Services Overwiev* [online]. 2010, aktualizováno 2010-05-08 [cit. 2011-01-08]. Dostupné na URL: <<http://social.technet.microsoft.com/wiki/contents/articles/active-directory-domain-services-ad-ds-overview.aspx>>
- [3] Čejka, R.: *Návrh, správa, bezpečnost: Virtuální privátní síť*. FIT VUT Brno, 2010.
- [4] *Remote Desktop Protocol* [online]. Wikipedia: 2011, aktualizováno 2010-12-14 [cit. 2011-01-07]. Dostupné na URL: <http://en.wikipedia.org/wiki/Remote_Desktop_Protocol>
- [5] *Proxy Server* [online]. Wikipedia: 2011, aktualizováno 2010-06-13 [cit. 2011-01-07]. Dostupné na URL: <http://en.wikipedia.org/wiki/Proxy_server>
- [6] Matoušek, P.: *Síťové aplikace a správa sítí: Systém DNS*. FIT VUT Brno, 2010.
- [7] Ščuglík, F.: *Přenos dat, počítačové sítě a protokoly: Struktura internetu a skupinové adresování*. FIT VUT Brno, 2008.
- [8] *Conectivity testing* [online]. MaximumAsp: 2006, aktualizováno 2009-05-26 [cit. 2011-01-07]. Dostupné na URL: <<https://help.maximumasp.com/KB/a445/connectivity-testing-with-ping-telnet-tracert-and-pathping-.aspx>>
- [9] *Help desk* [online]. Wikipedia: 2010, aktualizováno 2010-30-12 [cit. 2010-12-20]. Dostupné na URL: <http://en.wikipedia.org/wiki/Help_Desk>
- [10] Czegel, B.: *Running Effective Help Desk, 2nd Edition*. Wiley, New York, 1998. ISBN-13: 978-0471248163
- [11] Dvořák, J.: *Expertní systémy* [online]. VUT FSI Brno: 2004. [cit. 2010-12-20]. Dostupné na URL: <<http://www.uai.fme.vutbr.cz/~jdvorak/Opory/ExpertniSystemy.pdf>>

- [12] Kirkham, R: *SharePoint 2010 Architectures Overview* [online]. MSDN: 2011, aktualizováno 2011-03-09 [cit. 2011-04-05]. Dostupné na URL: <<http://msdn.microsoft.com/en-us/library/gg552610.aspx>>
- [13] *SharePoint Farm configuring and deployment* [online]. SP Magazine: 2009, aktualizováno 2011-03-03 [cit. 2011-04-06]. Dostupné na URL: <<http://sharepointmagazine.net/articles/best-practices-of-sharepoint-farm-configuring-and-deployment-part-1-architectural-and-logical-planning>>
- [14] *Compare SharePoint Editions* [online]. MS SharePoint: 2011, aktualizováno 2011 [cit. 2011-04-06]. Dostupné na URL: <<http://sharepoint.microsoft.com/en-us/buy/Pages/Editions-Comparison.aspx>>
- [15] *Asp.Net Web Forms* [online]. Asp.Net: 2010, aktualizováno 2010-06-02 [cit. 2011-04-08]. Dostupné na URL: <<http://www.asp.net/web-forms>>
- [16] Šeda, J: *J2EE, .NET a vývoj rozsáhlých systémů* [online]. Interval.cz: 2003, aktualizováno 2003-10-02 [cit. 2011-04-08]. Dostupné na URL: <<http://interval.cz/clanky/j2ee-net-a-vyvoj-rozsahlych-systemu-1>>
- [17] *Wamp server* [online]. 2010, aktualizováno 2010-12-24 [cit. 2011-04-04]. Dostupné na URL: <<http://www.wampserver.com/en/presentation.php>>
- [18] *Podrobnosti o licencích* [online]. SharePoint: 2011, aktualizováno 2011-01-01 [cit. 2011-03-28]. Dostupné na URL: <<http://sharepoint.microsoft.com/cs-cz/buy/Pages/Licensing-Details.aspx>>
- [19] *System requirements* [online]. MSDN: 2010, aktualizováno 2010-05-12 [cit. 2011-03-28]. Dostupné na URL: <<http://technet.microsoft.com/en-us/library/cc288955.aspx>>
- [20] *Events in SharePoint Foundation 2010* [online]. MSDN: 2010, aktualizováno 2010-05-01 [cit. 2011-03-29]. Dostupné na URL: <<http://msdn.microsoft.com/en-us/library/ms442323.aspx>>
- [21] *WebPart Class* [online]. MSDN: 2010, aktualizováno 2010-10-27 [cit. 2011-03-30]. Dostupné na URL: <<http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.webpartpages.webpart.aspx>>

- [22] *Examples of common formulas* [online]. MS Office: 2007, aktualizováno 2007-06-15 [cit. 2011-04-01]. Dostupné na URL: <<http://office.microsoft.com/en-au/windows-sharepoint-services-help/examples-of-common-formulas-HA010105479.aspx>>
- [23] *SPItemEventReceiver Class* [online]. MSDN: 2010, aktualizováno 2010-06-10 [cit. 2011-03-30]. Dostupné na URL: <<http://msdn.microsoft.com/en-us/library/ms462603.aspx>>
- [24] Macaalay, R.: *Developing Web Parts for Sharepoint 2010* [online]. Code Project: 2010, aktualizováno 2010-12-15 [cit. 2011-04-05]. Dostupné na URL: <<http://www.codeproject.com/Articles/136857/Developing-Web-Parts-for-Sharepoint-2010.aspx>>
- [25] Malina, P.: *Microsoft Windows Powershell: Podrobný průvodce skriptováním*. Computer Press, a.s., Brno, 2007. ISBN 978-80-251-1816-0
- [26] Havivi, A.: *SharePoint usage and statistics* [online]. Eggheadcafe: 2010, aktualizováno 2011-02-02 [cit. 2011-04-10]. Dostupné na URL: <<http://www.eggheadcafe.com/articlelist.aspx?topicid=69>>
- [27] Coventry, P.: *Microsoft SharePoint Designer 2010: Step by Step*. Microsoft Press, California, 2010. ISBN-13: 978-0-7356-2733-8
- [28] *Helpdesk Application* [online]. Jitbit, AC: 2011, aktualizováno 2011-01-02 [cit. 2011-01-10]. Dostupné na URL: <<http://www.jitbit.com/helpdesk-software>>
- [29] *Domain Trust* [online]. Microsoft TechNet: 2011, aktualizováno 2010-11-02 [cit. 2011-05-20]. Dostupné na URL: <<http://technet.microsoft.com/en-us/library/cc961481.aspx>>

Seznam příloh

Příloha 1. Obrázky

Příloha 2. Zdrojové kódy

Příloha 3. Manuály

Příloha 4. DVD

Příloha 1. Obrázky

Role Services: 41 installed

Role Service	Status
Web Server	Installed
Common HTTP Features	Installed
Static Content	Installed
Default Document	Installed
Directory Browsing	Installed
HTTP Errors	Installed
HTTP Redirection	Installed
WebDAV Publishing	Not installed
Application Development	Installed
ASP.NET	Installed
.NET Extensibility	Installed
ASP	Installed
CGI	Installed
ISAPI Extensions	Installed
ISAPI Filters	Installed
Server Side Includes	Installed
Health and Diagnostics	Installed
HTTP Logging	Installed
Logging Tools	Installed
Request Monitor	Installed
Tracing	Installed
Custom Logging	Not installed
ODBC Logging	Not installed
Security	Installed
Basic Authentication	Installed
Windows Authentication	Installed
Digest Authentication	Installed
Client Certificate Mapping Authentication	Installed
IIS Client Certificate Mapping Authentication	Installed
URL Authorization	Installed
Request Filtering	Installed
IP and Domain Restrictions	Installed
Performance	Installed
Static Content Compression	Installed
Dynamic Content Compression	Installed
Management Tools	Installed
IIS Management Console	Installed
IIS Management Scripts and Tools	Installed
Management Service	Installed
IIS 6 Management Compatibility	Installed
IIS 6 Metabase Compatibility	Installed
IIS 6 WMI Compatibility	Installed
IIS 6 Scripting Tools	Installed
IIS 6 Management Console	Installed
FTP Server	Not installed
FTP Service	Not installed
FTP Extensibility	Not installed
IIS Hostable Web Core	Not installed

Obrázek 7.1: Spuštěné služby web serveru (IIS).

Features: 5 of 42 installed

Group Policy Management
Message Queuing
Message Queuing Services
Message Queuing Server
Remote Server Administration Tools
Role Administration Tools
AD DS and AD LDS Tools
AD DS Tools
AD DS Snap-Ins and Command-Line Tools
Active Directory Administrative Center
Active Directory module for Windows PowerShell
DNS Server Tools
Web Server (IIS) Tools
Windows Process Activation Service
Process Model
.NET Environment
Configuration APIs
.NET Framework 3.5.1 Features
.NET Framework 3.5.1
WCF Activation
HTTP Activation
Non-HTTP Activation

Obr. 7.2: Spuštěné funkce na serveru.

Step 1

If Current Item:Problém equals Technické

Set Přiděleno to Technici

then Email Current Item:Created By: Technici

Else if Current Item:Problém equals Vývoj

Set Přiděleno to Programátoři

then Email Current Item:Created By: Programátoři

Else if Current Item:Problém equals Obchodní

Set Přiděleno to Obchod

then Email Current Item:Created By: Obchod

Else if Current Item:Problém equals Reklamace

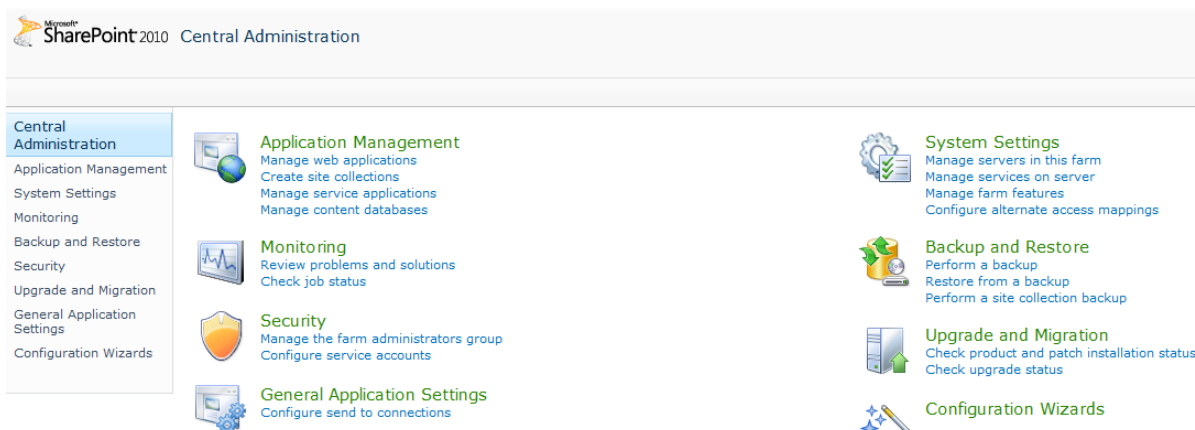
Set Přiděleno to Reklamace

then Email Current Item:Created By: Reklamace

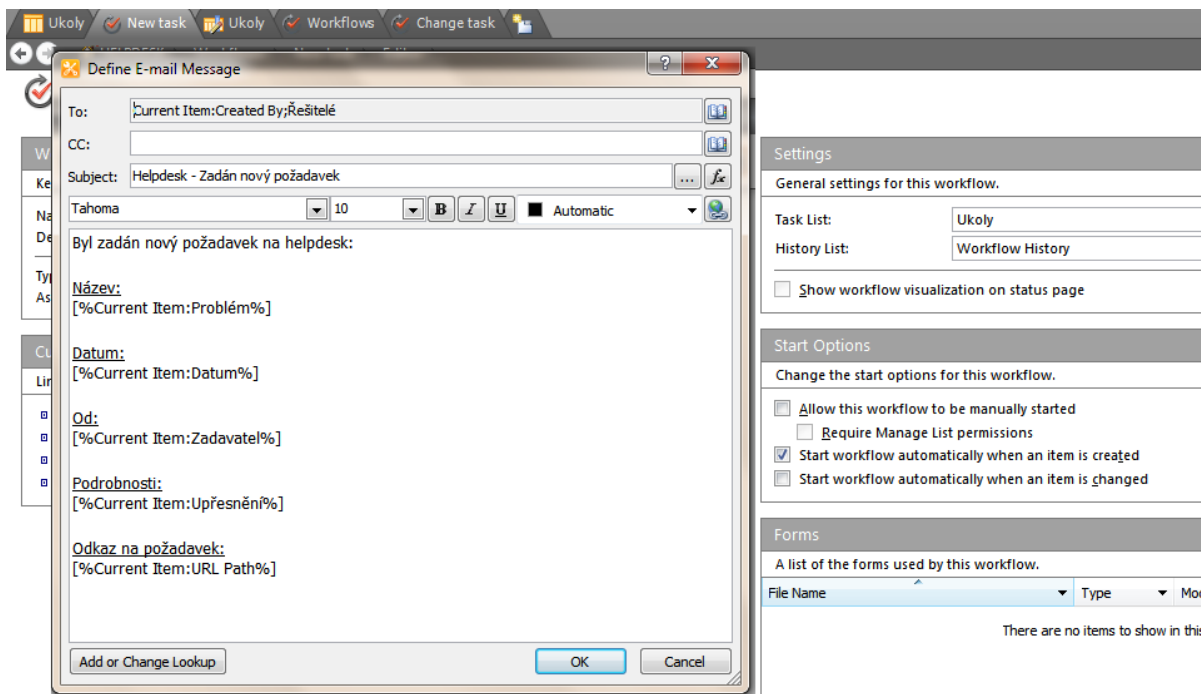
Obrázek 7.6: Editace jednoho kroku pracovního postupu.

Server: SP-SERVER View: Configurable		
Service	Status	Action
Business Data Connectivity Service	Started	Stop
Central Administration	Started	Stop
Claims to Windows Token Service	Stopped	Start
Microsoft SharePoint Foundation Incoming E-Mail	Started	Stop
Microsoft SharePoint Foundation Sandboxed Code Service	Stopped	Start
Microsoft SharePoint Foundation Subscription Settings Service	Stopped	Start
Microsoft SharePoint Foundation Web Application	Started	Stop
Microsoft SharePoint Foundation Workflow Timer Service	Started	Stop
SharePoint Foundation Help Search	Started	Stop

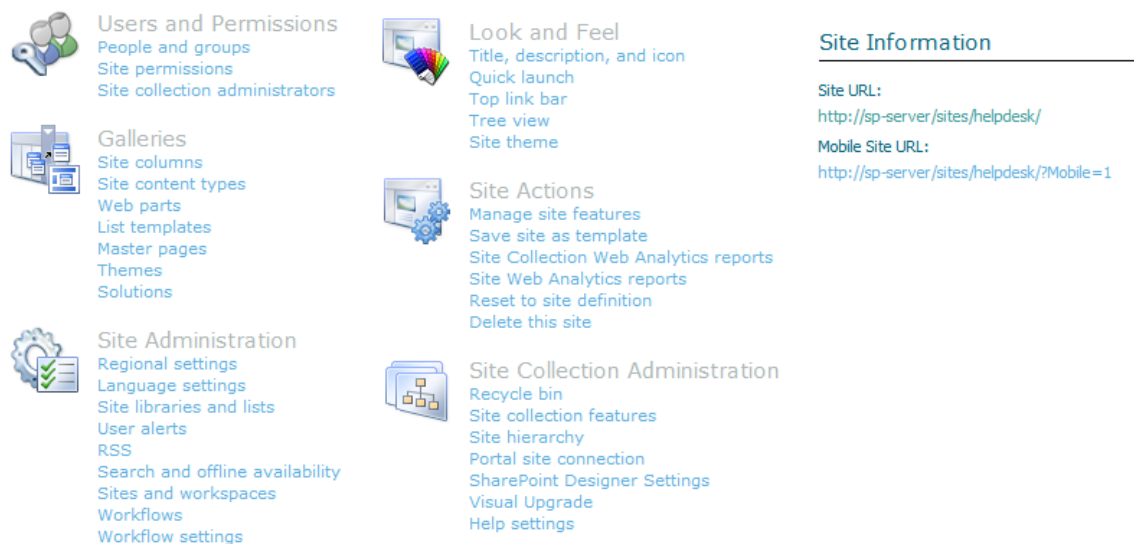
Obrázek 7.3: Spuštěné SharPoint Foundation služby na serveru.



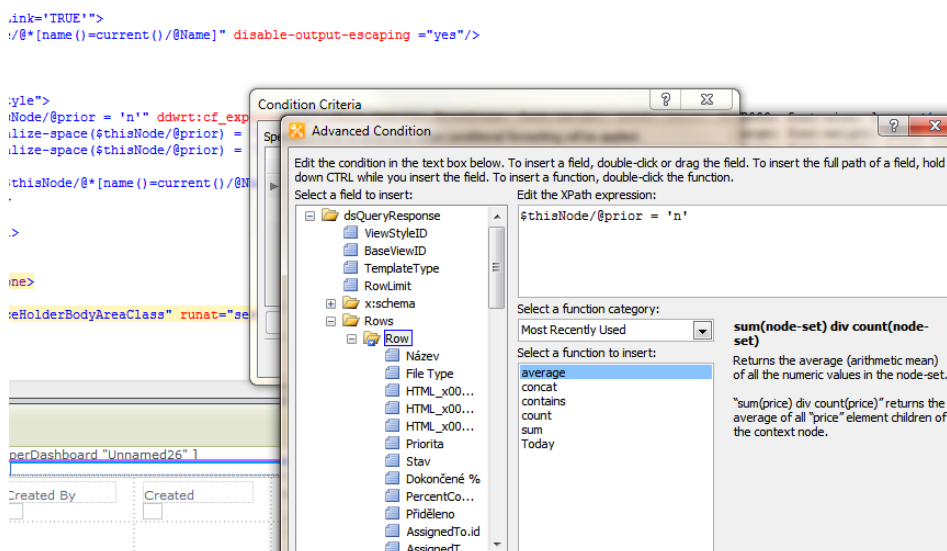
Obrázek 7.4: SharePoint Central Administration.



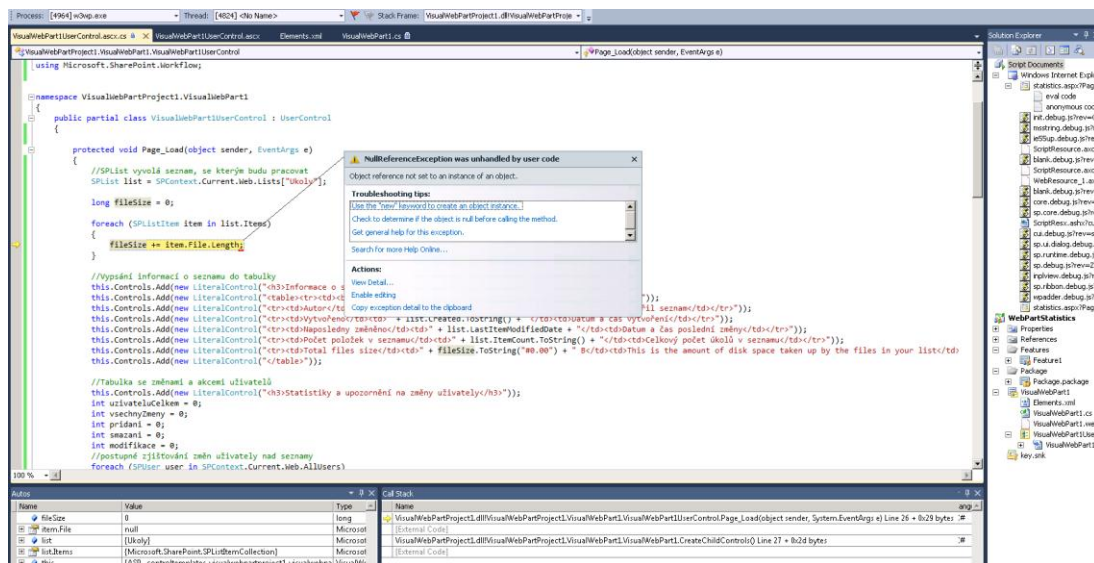
Obrázek 7.5: Akce zaslání mailu při vytvoření požadavku.



Obrázek 7.7: Možnosti nastavení vytvořené SharePoint stránky přímo v prohlížeči.



Obrázek 7.8: Možnost nastavení podmíněných kritérií.



Obrázek 7.9: Ladění webové části ve Visual Studiu 2010.


Search this site...

Posted By

Post

Started: 5/2/2011 6:52 PM

View Properties | Reply




SP\administrator

Otázka ohledně funkcí aplikace
Chci se zeptat zda je možné.....

Posted: 5/2/2011 6:53 PM


View Properties | Reply




SP\administrator

Je možné problém vyřešit následovně....

Show Quoted Messages

Type	Name	Modified By	Modified	Created By	Created
	Nastavení programu	SP\administrator	5/2/2011 6:59 PM	SP\administrator	5/2/2011 6:57 PM


Add new page

Type	Name	Modified	Modified By
	obnova spojení při výpadku	5/2/2011 6:44 PM	SP\administrator


Add document

New | Upload | Actions | Settings

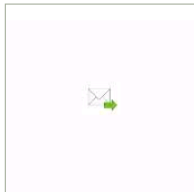
View: All Pictures



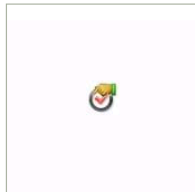
bns



ok



send



test

Obrázek 7.10: Ukázka zobrazení dokumentů a knihoven v helpdesku
(konkrétně diskuze, wiki page, knihovna dokumentů a obrázků).

Příloha 2. Zdrojové kódy

Zdrojové kódy souborů projektu webové části ve Visual Studiu 2010:

VisualWebPart.cs

```
using System;
using System.ComponentModel;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using Microsoft.SharePoint;
using Microsoft.SharePoint.WebControls;

namespace VisualWebPartProject1.VisualWebPart1
{
    [ToolboxItemAttribute(false)]
    public class VisualWebPart1 : WebPart
    {
        // Visual Studio might automatically update this path when you change the Visual Web Part
        item
        private const string _ascxPath =
@"~/_CONTROLTEMPLATES/VisualWebPartProject1/VisualWebPart1/VisualWebPart1UserControl.ascx";
        protected override void CreateChildControls()
        {
            Control control = Page.LoadControl(_ascxPath);
            Controls.Add(control);
        }
    }
}
```

VisualWebPart1UserControl.ascx.cs

```
using System;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Security.Permissions;
using Microsoft.SharePoint;
using Microsoft.SharePoint.Security;
using Microsoft.SharePoint.Utilities;
using Microsoft.SharePoint.Workflow;

namespace VisualWebPartProject1.VisualWebPart1
{
    public partial class VisualWebPart1UserControl : UserControl
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            //SPList vyvolá seznam, se kterým budu pracovat, využiji přitom objekt SPList
            SPList list = SPContext.Current.Web.Lists["Ukoly"];

            //Vypsání potřebných informací o daném seznamu do tabulky
            this.Controls.Add(new LiteralControl("<h3>Informace o seznamu úkolů</h3>"));
            this.Controls.Add(new LiteralControl
            ("<table><tr><td><b>Vlastník</b></td><td><b>Info</b></td><td><b>Popis</b></td></tr>"));
            ;
            this.Controls.Add(new LiteralControl
            ("<tr><td>Autor</td><td>" + list.Author.LoginName +
            "</td><td>Uživatel, který vytvořil seznam</td></tr>"));
            this.Controls.Add(new LiteralControl
            ("<tr><td>Vytvořeno</td><td>" + list.Created.ToString() +
            "</td><td>Datum a čas vytvoření</td></tr>"));
            this.Controls.Add(new LiteralControl
            ("<tr><td>Naposledy změněno</td><td>" + list.LastItemModifiedDate +
            "</td><td>Datum a čas poslední změny</td></tr>"));
            this.Controls.Add(new LiteralControl
            ("<tr><td>Počet položek v seznamu</td><td>" + list.ItemCount.ToString() +
            "</td><td>Celkový počet úkolů v seznamu</td></tr>"));
            this.Controls.Add(new LiteralControl("</table>"));
        }
    }
}
```

Element.xml

```
<?xml version="1.0" encoding="utf-8"?>
<Elements xmlns="http://schemas.microsoft.com/sharepoint/" >
  <Module Name="VisualWebPart1" List="113" Url="catalogs/wp">
    <File Path="VisualWebPart1\VisualWebPart1.webpart"
    Url="VisualWebPartProject1\VisualWebPart1.webpart" Type="GhostableInLibrary" >
      <Property Name="Group" Value="Custom" />
    </File>
  </Module>
</Elements>
```

VisualWebPart1UserControl.ascx

```
<%@ Assembly Name="$SharePoint.Project.AssemblyFullName$" %>
<%@ Assembly Name="Microsoft.Web.CommandUI, Version=14.0.0.0, Culture=neutral,
PublicKeyToken=71e9bce111e9429c" %>
<%@ Register Tagprefix="SharePoint" Namespace="Microsoft.SharePoint.WebControls"
Assembly="Microsoft.SharePoint, Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c" %>
<%@ Register Tagprefix="Utilities" Namespace="Microsoft.SharePoint.Utilities"
Assembly="Microsoft.SharePoint, Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c" %>
<%@ Register Tagprefix="asp" Namespace="System.Web.UI" Assembly="System.Web.Extensions, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=31bf3856ad364e35" %>
<%@ Import Namespace="Microsoft.SharePoint" %>
<%@ Register Tagprefix="WebPartPages" Namespace="Microsoft.SharePoint.WebPartPages"
Assembly="Microsoft.SharePoint, Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c" %>
<%@ Control Language="C#" AutoEventWireup="true" CodeBehind="VisualWebPart1UserControl.ascx.cs"
Inherits="VisualWebPartProject1.VisualWebPart1.VisualWebPart1UserControl" %>
```

Event Receiver pro změnu uživatelských práv nad položkami seznamu:

```
using System;
using System.Security.Permissions;
using Microsoft.SharePoint;
using Microsoft.SharePoint.Security;
using Microsoft.SharePoint.Utilities;
using Microsoft.SharePoint.Workflow;

namespace MyEventReceiver.EventReceiver1
{
    /// <summary>
    /// List Item Events
    /// </summary>
    public class EventReceiver1 : SPItemEventReceiver
    {
        public void permissions(SPItemEventProperties properties)
        {
            //SPListItem určuje položku v seznamu úkolů, na kterou byla událost vyvolána
            SPListItem item = properties.ListItem;
            item.ResetRoleInheritance();
            item.BreakRoleInheritance(false);
            //vynulování a zrušení dědění uživatelských práv na daný seznam

            //identifikace zadavatele - uživatele, který vložil úkol
            string creator = item["Created By"].ToString();
            int zadavatel = Convert.ToInt32(creator.ToString().Substring(0, creator.IndexOf(';')));

            //zjištění řešitele - uživatel určen pro řešení úkolu
            string prideleno = item["Přiděleno"].ToString();
            int resitel = Convert.ToInt32(prideleno.ToString().Substring(0, prideleno.IndexOf(';')));

            //vytvoření definic rolí na vytvořené položce - úkolu
            SPRoleDefinition read = properties.OpenWeb().RoleDefinitions["Read"];
            SPRoleDefinition contrib = properties.OpenWeb().RoleDefinitions["Contribute"];
            SPRoleDefinition admin = properties.OpenWeb().RoleDefinitions["Full Control"];

            //vytvoření uživatelských rolí na vytvořené položce
            SPRoleAssignment zadavatel_role = new SPRoleAssignment(properties.OpenWeb().Users.GetByID(zadavatel));
            //zadavatel
            SPRoleAssignment resitel_role = new SPRoleAssignment(properties.OpenWeb().Users.GetByID(resitel));
            //řešitel
            SPRoleAssignment ostatni_role = new SPRoleAssignment(properties.OpenWeb().Groups.GetByID(8));
            //ostatní uživatelé
            SPRoleAssignment admin_role = new SPRoleAssignment(properties.OpenWeb().Groups.GetByID(2));
            //administrátoři

            //přiřazení vytvořených definic k uživatelským rolím
            zadavatel_role.RoleDefinitionBindings.Add(contrib);
            resitel_role.RoleDefinitionBindings.Add(contrib);
        }
    }
}
```

```

        ostatni_role.RoleDefinitionBindings.Add(read);
        admin_role.RoleDefinitionBindings.Add(admin);

        //vlození rolí do položky
        item.RoleAssignments.Add(zadavatel_role);
        item.RoleAssignments.Add(resitel_role);
        item.RoleAssignments.Add(ostatni_role);
        item.RoleAssignments.Add(admin_role);

        item.Update();
    }

    /// <summary>
    /// An item was added.
    /// </summary>
    public override void ItemAdded(SPIItemEventProperties properties)
    {
        this.permissions(properties);
    }

    /// <summary>
    /// An item was updated.
    /// </summary>
    public override void ItemUpdated(SPIItemEventProperties properties)
    {
        this.permissions(properties);
    }
}
}

```

Část pracovního postupu přidělení řešitele v aspx kódu:

```

<xsl:template name="dvt_1">
    <xsl:variable name="dvt_StyleName">RepForm3</xsl:variable>
    <xsl:variable name="Rows" select="/dsQueryResponse/NewDataSet/Row"/>
    <xsl:variable name="dvt_RowCount" select="count($Rows)" />
    <xsl:variable name="RowLimit" select="1" />
    <xsl:variable name="FirstRow" select="$dvt_firstrow" />
    <xsl:variable name="LastRow">
        <xsl:choose>
            <xsl:when test="($FirstRow + $RowLimit - 1) > $dvt_RowCount"><xsl:value-of select="$dvt_RowCount" /></xsl:when>
            <xsl:otherwise><xsl:value-of select="$FirstRow + $RowLimit - 1" />
            </xsl:otherwise>
        </xsl:choose>
    </xsl:variable>
    <xsl:variable name="IsEmpty" select="$dvt_RowCount = 0" />
    <xsl:call-template name="dvt_1.header">
        <xsl:with-param name="Rows" select="$Rows" />
    </xsl:call-template>
    <table border="0" width="100%">
        <xsl:call-template name="dvt_1.body">
            <xsl:with-param name="Rows" select="$Rows[position() >= $FirstRow and position() <= $LastRow]"/>
            <xsl:with-param name="FirstRow" select="1" />
            <xsl:with-param name="LastRow" select="$dvt_RowCount" />
        </xsl:call-template>
    </table>
    <xsl:call-template name="dvt_1.commandfooter">
        <xsl:with-param name="FirstRow" select="$FirstRow" />
        <xsl:with-param name="LastRow" select="$LastRow" />
        <xsl:with-param name="RowLimit" select="$RowLimit" />
        <xsl:with-param name="dvt_RowCount" select="$dvt_RowCount" />
        <xsl:with-param name="RealLastRow" select="number(ddwrt:NameChanged('','-100'))" />
    </xsl:call-template></xsl:template>

```


Skript v PowerShellu pro kontrolu spojení:

```
param(
    #Definice parametrů
    [string] $Firma = "Delikomat",
    [string] $Problem = "Technické",
    [string] $Detail = "Mail",
    [string] $remoteHost = "proxy.delikomat.cz",
    [int] $port = 3389,
    [int] $commandDelay = 1000
)

function Main
{
    #pokračuje i přes chyby (při spojení)
    $ErrorActionPreference = "SilentlyContinue"
    # Otevře socket a připojí se k hostovi na specifickém portu
    # Pokud dojde k chybě nebo timeoutu, pokračuje dál (zapsání neúspěchu)
    trap {continue} $socket = new-object System.Net.Sockets.TcpClient($remoteHost, $port)
    #write-host "Connecting to $remoteHost on port $port"

    #pripojeni k sharepoint webu
    Add-PsSnapin Microsoft.SharePoint.PowerShell -ErrorVariable err -ErrorAction
        SilentlyContinue
    [System.Reflection.Assembly]::LoadWithPartialName("Microsoft.SharePoint")
    $spsite=new-object Microsoft.SharePoint.SPSite("http://sp-server/sites/helpdesk/")

    #otevření webu
    $spweb=$spsite.OpenWeb()
    $spAssignment
    #získání potřebného seznamu
    $SPList = $spweb.Lists["Ukoly"]
    #určení konkrétní položky
    $SPItem = $SPList.Items | Where { $_["Firma"] -eq $Firma -and $_["Problém"] -eq $Problem -
        and $_["Upřesnění"] -eq $Detail }

    #pokud se v časovém intervalu nespojí
    If($socket -eq $null)
    {
        #doplň do dané položky tuto informaci
        $SPItem | ForEach { $_["Test"] = "Not OK"; $_.Update() }
        Write-Error "Could not connect to remote computer: $_"
        $SPAssignment
    }

    #pokud spojení proběhne v pořádku
    Else{
        #doplň do dané položky tuto informaci
        $SPItem | ForEach { $_["Test"] = "OK"; $_.Update() }
        write-host "Connected.`n"
        $SPAssignment
    }
}

. Main
```

Možnost nastavení zobrazení ve formulářích v PowerShellu:

```
function Main
{
    #pripojeni k sharepoint webu
    Add-PsSnapin Microsoft.SharePoint.PowerShell -ErrorVariable err -ErrorAction
        SilentlyContinue
    [System.Reflection.Assembly]::LoadWithPartialName("Microsoft.SharePoint")
    $spsite=new-object Microsoft.SharePoint.SPSite("http://sp-server/sites/helpdesk/")

    #otevření webu
    $spweb=$spsite.OpenWeb()
    #získání potřebného seznamu
    $SPList = $spweb.Lists["Ukoly"]
    $SPField = $SPList.Fields["Test"]
    #zamezení zobrazení v zadání nového úkolu
    $SPField.ShowInNewForm = 0;
    $SPField.Update()
    $spweb.Dispose()
    $spsite.Dispose()
}
. Main
```

Příloha 3. Manuály

Instalace a nastavení serveru pro zprovoznění SharePoint helpdesku

- 1) Instalace serveru s názvem **SP-SERVER** a OS Windows Server 2008 R2.
- 2) Přidělit serveru potřebné role, spustit funkce dle *obr. 7.1* (webový, aplikační server).
- 3) Přidání serveru do domény **SP.local**, vytvořit Active Directory a v něm tyto účty:
Administrátorské účty - *SQL_Admin, SP_Admin, SP_Service, SP_Search* a účty uživatelů dle tabulky
- 4) Instalace SQL Server 2008 R2
 - defaultní nastavení a cesty, instalace database services
 - vytvořený server pojmenovat **SQL_SP**, jako administrátora uživatele *SQL_Admin* (windows autentizace)
 - spuštění služeb SQL Server, SQL Server Agent
- 5) Instalace SharePoint Foundation 2010
 - instalace podpůrného balíčku (obsaženo)
 - stand-alone instalace, umístění dat na C:\SharePoint
 - přidání uživatele *SP_Admin* jako administrátora
 - připojení na databázový server **SQL_SP**
 - vybrat porty (80,81), NTLM autentizace
 - vytvořit webovou aplikaci **sp-server.sp.local**
 - umístit kolekci webů na *http://sp-server/sites/helpdesk*.
 - přidat administrátory pro služby – uživatelé *SP_Admin, SP_Search, SP_Service*
- 6) Konfigurace v SharePoint 2010 Central Administration
 - spuštění služeb dle obrázku 7.3
 - konfigurace SMTP serveru.
- 7) Instalace vývojových nástrojů
 - SharePoint Designer 2010 - připojení na vytvořenou kolekci webů, naplnění uživateli z AD
 - Visual Studio 2010 – vytvoření dalších částí dle šablon
- 8) Obnova webové farmy ze zálohy a zprovoznění všech funkcí v SharePoint Management Shell
 - Obnova zálohy v *SharePoint 2010 Central Administration* nebo pomocí příkazu
Restore-SPSite -Identity http://sp-server/sites/helpdesk -Path "c:\backup.bak"
 - doplnění vytvořených Event Receiverů do galerie řešení ze souboru wsp
Add-SPSolution C:\EventReceiver.wsp
Install-SPSolution EventReceiver.wsp -GACDeployment
 - doplnění vytvořených webových částí do galerie řešení ze souboru wsp
Add-SPSolution C:\WebPart.wsp
Install-SPSolution -identity WebPart.wsp -WebApplication
http://sp-server/sites/helpdesk - GACDeployment
 - umístění všech testovacích skriptů do C:\Scripts

Příloha 4. DVD

Obsahem média, které je přiloženo k práci, jsou jednak zdrojové kódy programu, včetně všech souborů, záloh aplikace a podrobného návodu pro spuštění aplikace. Dále obsahuje i tento dokument v různých formátech a některé použité programy.

Struktura obsahu DVD:

Dokumentace

Text diplomové práce ve formátech *doc*, *docx* a *pdf* a manuál ke zprovoznění aplikace.

Helpdesk

Zdrojové kódy, vytvořené části a zálohy celé aplikace.

- **Backups** - záloha kolekce webu a celé webové farmy.
- **Solutions** - *wsp* soubory pro nasazení webových částí a event receiverů.
- **Source** - zdrojové kódy celého vytvořeného SharePoint webu, Event Receiverů, webových částí a PowerShell skriptů.

Programy

Programy pro zprovoznění a testování helpdesku. Obsahuje instalační soubory programů: SharePoint Foundation 2010, SharePoint Designer 2010, SQL Server 2008 R2 Express.

README.txt - podrobný popis obsahu DVD.